

世纪互联蓝云研究院丛书

# Microsoft Azure 管理与开发（上册）

## 基础设施服务 IaaS

世纪互联蓝云公司 主编

韩旭 张立鹤 左滕 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是当下关于 Microsoft Azure 产品的书籍中特别具有完整性、实用性的一本书，由 Microsoft Azure 中国区运维团队——世纪互联蓝云的资深工程师们编写。本书内容贴合实际，整合了运维团队在处理客户问题过程中积累的大量经验和案例，汇总了大量的解决方案，操作方法，内容深入浅出，可操作性极强。

本书内容完整覆盖了 Microsoft Azure 产品中 IaaS 各个方面的内容，主要包括计算节点，存储资源，虚拟网络，安全配置，负载均衡架构设计，高可用架构设计，备份与还原，内容分发网络，自动化运维，Azure 活动目录，常见排错方法等，针对原理做了深入的解析，并结合大量实例将原理与实践相结合。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目（CIP）数据

Microsoft Azure 管理与开发. 上册, 基础设施服务 IaaS / 世纪互联蓝云公司主编; 韩旭等编著. —北京: 电子工业出版社, 2017.9

(世纪互联蓝云研究院丛书)

ISBN 978-7-121-32649-3

I. ①M… II. ①世… ②韩… III. ①云计算 IV. ①TP393.027

中国版本图书馆 CIP 数据核字 (2017) 第 218401 号

策划编辑: 张瑞喜

责任编辑: 张瑞喜

印 刷: 中国电影出版社印刷厂

装 订: 中国电影出版社印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 26 字数: 633 千字

版 次: 2017 年 9 月第 1 版

印 次: 2017 年 9 月第 1 次印刷

定 价: 78.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式: [zhangruixi@phei.com.cn](mailto:zhangruixi@phei.com.cn)。

# 序 言

## “大运维”引领中国云计算的进阶之路

——世纪互联蓝云总裁 柯文达

“这是最好的时代，也是最坏的时代。”这个时代一切都是那样欣欣向荣、高速发展，但是高速发展的同时也带来阵痛与焦虑。去年世纪互联蓝云出版第一本书《Office 365 管理员实战指南》的时候，云计算在中国还尚处于落地的阶段，并呈现百家争鸣的态势；而今年，正值世纪互联蓝云为 Azure 继续出书之际，云计算在中国这片土地上也已经落地生根，在不断蜕变的同时，与更多新兴行业一同前进、迅速发展。世纪互联蓝云有幸一直走在新技术的前沿，也有幸结识了不少精诚合作的伙伴，并与中国的云计算一起走在不断进阶的路上。



### 砥砺前行，打造世界一流本土可信云服务

今年是世纪互联蓝云成立 4 周年，也是微软智能云入华 3 周年的日子。在这 4 年多的时间里，世纪互联与微软达成了长期默契的合作伙伴关系，并把国际一流水准的云服务深深扎根中国市场；经过 3 年多可靠安全的运营，中国用户也越来越信赖由世纪互联运营的微软云服务。由世纪互联运营的云计算用户获得了迅猛增长：Azure 企业客户从最初的 5 万家增长到 8 万家，Office 365 企业用户从 3.5 万增长到 120 万，在中国云计算行业中的用户增长成绩是非常突出的。

另外使用由世纪互联运营的云服务客户也覆盖了教育、互联网、医疗、政府、金融、汽车、AI、媒体等全行业，表明云计算已经被中国大多数行业的企业所接受并使用。其中既有观致汽车、蒙牛集团、三星这样三年来始终支持 Azure 的老客户，也有微鲸电视、棠棣科技、白山云等人工智能新行业的企业。使用由世纪互联运营的云服务的客户在中国 top10 的城市分布：北京、上海、深圳、广州、杭州、成都、南京、重庆、苏州、天津。与 2016 年中国 GDP 最高的 10 个城市基本吻合，也证明云服务使用率最高的地区主要在经济发达的城市。

经过 3 年多发展壮大，世纪互联的云运维团队也从最初的几十人，增长到了今天的 400 多人，成为中国最大的专业云运维团队之一；三年来总接听电话数 220 000，总接听电话时长 500 000 分钟；三年来共解决用户问题 150 000 个；平均每年接听电话数和解决用户问题数增长 50% 以上。用户问题的逐年增长，证明中国用户使用云计算服务已经进入到了深入阶段；三年共完成 80 527 个部署，其中 Azure 完成 66 728 个部署，Office 365 完成 13 799 个部署。世纪互联工程师在三年多的运营过程中具备了丰富的经验，并总结出了一系列技术相关文章，三年来运维团队共撰写技术文章 13 800 多页。此外，世纪互联的云运维团队还以做云计算行业的“七星级酒店”的服务为定位，建立一套科学高效、安全合规的标准

化的服务流程规范，形成了独有的云运维体系，确保了为用户提供高标准、高质量、高可靠的云服务。虽然 Azure 和 Office 365 用户的不断增长带来了运维难度和复杂程度的增加，但是世纪互联的工程师在高强度的压力下，圆满完成任务，三年来用户满意度达到 95% 以上，大大高于云计算行业平均水平的 70%。

### 拥抱未来新趋势，积极服务中国用户

经过多年的深入发展，云计算在今年也逐渐呈现了一些新的趋势，而世纪互联蓝云一直以来都以前瞻的视野，积极拥抱新趋势，希望与合作伙伴一起更好地服务中国用户。

**云加 AI 成为新的趋势：**作为提升数据处理效率的有利武器，AI 需要找到适合自己的落脚点，云天然地是数据处理的天然土壤。云端的海量数据，给人工智能提供了宝贵的训练数据，已经成为云服务巨头们极为看重的市场，也是云计算竞争进一步升级的领域。

**混合云是用户首选：**云是企业数字化转型必不可少的基石，越来越多的行业用户正将其业务和数据迁移到云上。而借助混合云，用户既可以发挥公有云低成本、可扩展等优势，又可以通过私有云保护重要的数据。IDC 在 2016 年发布的报告显示，从成本优化和灵活性、安全性、合规性等方面综合考虑，73% 的企业用户部署了混合云方案。

**单独的 IaaS 越来越同质化，需要积极发展 PaaS 服务：**最近几年 IaaS 和 SaaS 保持高速增长，相对而言 PaaS 份额较少。随着 IaaS 市场发展，产品越来越趋向同质化，但随着企业应用市场的发展和差异化竞争的需要，云计算运营商需要发展 PaaS 来满足客户需求和差异化竞争，根据行业客户需求或场景化需求来发展 PaaS 业务是未来的发展方向。

世纪互联蓝云今年以来在以上三个方面也做出了积极的布局和响应：一方面，世纪互联与微软已经加紧了混合云解决方案的落地，希望通过 Azure stack 混合云方案打通线上和线下能力，助力中国用户使用混合云实现更完善的云应用；另一方面，微软正在积极布局 AI，将 AI 整合到云计算的产品和服务中，世纪互联蓝云也将投入更多云运维能力，助力云计算行业为用户提供高效利器；此外，在 PaaS 业务方面，微软也不断研发更多贴近用户、带来前瞻能力的创新功能等，世纪互联蓝云也逐步将这些功能第一时间引入中国市场。世纪互联蓝云希望与合作伙伴共同拥抱新趋势，给中国用户带来更多裨益。

### 写在最后

为《Microsoft Azure 管理与开发（上册）基础设施服务 IaaS》写这篇序的时候，适逢世纪互联蓝云刚刚凭借 Azure 混合云解决方案通过了业界首次评出的“混合云解决方案评估”，同时获得了“可信云技术创新奖——混合云奖”。这是世纪互联蓝云获得的第 10 个可信云认证，也是可信云大会评出的第一个混合云方面的认证和奖项。如此殊荣的获得，是行业 and 用户对蓝云的厚爱，同时也证明世纪互联蓝云在公有云领域走在了前列，在混合云解决方案的能力上也受到了行业的认可。我为蓝云这样一支精进而专业的团队感到自豪，同时非常感谢我的团队这几年来的辛勤付出。

最后，我很高兴地看到蓝云的技术工程师们陆续把自己最前沿的技术和最具实操性的经验编撰成书，为业界 and 用户提供些许参考。如果这本书恰好可以作为您的借鉴，抑或我们的抛砖引玉引起了您更多思考、甚至是创新火花的迸发，那将是我们极大的荣幸。这也将激励我们更加努力前行，与云计算行业的同仁们共同开创新的进阶之路。

2017 年 8 月



# 前言

小时候，家里是没有电的，母亲经常在一盏昏暗的油灯下缝缝补补。夏日的夜晚，小孩子们都是在月光下追打嬉闹。村里也是没有自来水的，村子里只有三眼水井，一个轱辘横在井台上，手摇轱辘，带动水桶起起落落提取井水，村民的一切用水都靠着这些个轱辘和井提供。村民们日出而作，日落而息。这种唯美的田园生活大概已经传承了几百年，不曾改变过。

而今三十多年过去，村庄已经不见了，高楼大厦拔地而起，变成了城市的一部分。自来水、电、煤气等基础设施已经被视为理所当然；电视、汽车、智能手机、计算机也已是寻常之物；网上支付、顺风车、共享单车等新生事物层出不穷，甚至远在大山里的少数民族也已拥有了网约巴士服务……

那么，未来三十年又是什么样子呢？……

历史可以告诉未来。打开人类的文明史，我们不难发现：能够极大地改进人类生产方式、改善人类生活、推动人类文明进步的核心力量是科学和技术的革命，尤其是最近一百多年来的四次工业革命。与前几次工业革命不同，第四次工业革命以数字化转型为代表，结合了机器人、物联网、生物工程以及新能源等各种各样的技术，呈现出指数级而非线性的发展速度，势必将给我们的社会、经济和个人带来前所未有的改变，势必会让所有的行业都面临着变革与升级，势必会对我们生产和生活的方方面面造成冲击。一场浩浩荡荡的数字化革命已经势不可挡。李克强总理在 2017 年大连夏季达沃斯论坛开幕式致辞中指出，与以往的工业革命相比，在新一轮工业革命中实现包容性增长，具有更大的可能性，因为以网络化、数字化、智能化为代表的新一轮工业革命，不仅创造了新的供给和需求，大大拓展了发展空间，也给各方带来更多平等参与的机会，几乎每个人都可以借助互联网，更加便利地创业、创新、创富。新一轮工业革命将让更多人有了改变自身命运的机会，所有人都将有机会站在新时代的潮头上。

新一轮工业革命的基础和动力是云计算，新一轮工业革命将产生大量的数据，而这些数据需要经过大量的计算才能转变为应用。人们经常把云计算比作数字化时代的水、煤、电。参考美国国家标准与技术研究院（NIST）定义：云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络，服务器，存储，应用软件，服务），这些资源能够被快速提供，只需投入很少的管

理工作，或服务供应商进行很少的交互。云计算按照提供的服务层次可以分为 IaaS（基础设施即服务）、PaaS（平台即服务）和 SaaS（软件即服务）三种服务类型。云计算和水、煤、电之间的相似性很高。云计算提供的是一个诸如自来水网、煤气管网、电网一样的新的计算网络，用户只需要简单配置一下接口（如同安装一个水龙头，接一个电源插座一样），就可以通过通信端口和互联网方便地使用各种数据服务。

云计算搭建了一个崭新的平台，它将基础设施、硬件、软件、网络带宽、操作系统、数据库、中间件、平台技术等融合在一起，通过一个标准的模式，向最终用户提供终端云计算服务。用户能像用水、煤、电一样使用云计算服务，降低了使用 IT 服务的（包括金钱、人力、技术、效率等方面）诸多门槛。站在云计算的平台之上，充分利用云计算的高可靠性、高可扩展性、按需服务、价格低廉等特点，无论是政府用户、企业用户，还是个人用户，都可以更加专注于自己的业务，并且与自己行业的大数据、新技术、新流程、新管理手段相结合，就能够演变出更加丰富多彩的创新应用。

在风起云涌的云计算行业，微软和世纪互联毫无疑问是行业内的领导者。首个在中国正式商用的国际公有云，是由世纪互联运营的微软云。上海蓝云网络科技有限公司（简称：世纪互联蓝云）是世纪互联成立的专门运营 Microsoft Azure，Office 365 和 Power BI 等云服务的全资子公司，400 多人的国际水准云计算运营和服务团队，为客户提供包括 IaaS、PaaS、SaaS 在内的全方位云服务。世纪互联蓝云是中国第一家拥有国际一流技术水平的国内云计算服务提供商，拥有首屈一指的国际本土化运维服务能力，以及国内最有安全保障的、可靠的云服务。

由世纪互联运营的 Microsoft Azure 是一个不断增长的云服务平台，开发人员和 IT 专业人员通过设置在北京和上海的数据中心网络来构建、部署和管理应用程序。使用 Microsoft Azure，您可以自由地利用您所选择的工具、程序和框架等在所需的任何位置构建和部署您的任何应用。2013 年 5 月，由世纪互联运营的 Microsoft Azure 在中国开启公众预览，次年，正式商用。截至 2017 年初，由世纪互联运营的 Microsoft Azure 的云计算规模已经扩展了一倍，积累了 80 000 多家企业用户；功能也在不断丰富，新推出了 60 多种云服务；世纪互联蓝云的技术实力也在不断地积累，帮助客户解决了 150 000 多个技术难题，总结出了数千篇技术文档。这数千篇技术文档是数百位工程师辛苦劳动的结晶，是世纪互联蓝云的宝贵财富。

本书是世纪互联蓝云技术支持工程师多年来实践经验的总结，是从数千篇技术文档中精挑细选出来的精华，整合了团队在处理客户问题过程中的大量案例，汇总成系列的解决方案和最佳实践，针对有些方案还给出了具体的代码示例。其主要内容集中在由世纪互联运营的 Microsoft Azure 的技术、部署、应用、管理和开发等方面。由于由世纪互联运营的 Microsoft Azure 的功能众多、技术宽广、内容丰富，所以本书分为上、下两册，上册主要介绍 IaaS 服务，下册主要论述 PaaS 服务。

如今，基于 Microsoft Azure 的各种新应用层出不穷，对于致力于云计算技术的读者来说，必须不断加强学习，才能跟上新一轮科技革命和产业变革的步伐。本书的目的是希望能加深读者对 Microsoft Azure 的了解，希望能帮助企业（或组织）的 IT 管理员更好地掌握 Microsoft Azure 的部署和管理，能帮助开发人员设计出更好的产品和技术方案，能帮助技术爱好者更深入地理解 Microsoft Azure 的技术美感。

本书是一本实践书籍，阅读本书之前，您最好已经有了一个由世纪互联运营的 Microsoft Azure 账户，以方便您边看书、边实践。如果您还没有这样一个账户，请登录官方网站 [www.azure.cn](http://www.azure.cn)，只需花几分钟的时间，就可以拥有一个全新的 Azure 账户。

本书是世纪互联蓝云技术支持中心工程师们利用业余时间集体创作的结果，除了主要作者，还有很多工程师参与了写作，也有很多专家给出了具体的指导建议。参加本书（上册）编写的作者有：左滕、张立鹤、韩旭、贺俐铭、王为、边明凯、陆一通、胡明月、李小兵、傅张良、李振山、李海涛、王瑞丰、李岳、张璋、周坤、孙剑、张涛、王剑锋、艾文侠、许金金、李建新、杨田、谭巴莽、孔德路等。

非常感谢我们的同事孟钊宇女士和刘志兵先生，在他们的热情帮助下，本书才得以出版。

由于时间和技术水平有限，书中难免会出现一些错误或者不准确的地方，恳请您批评指正。同时，由于 Microsoft Azure 产品内容越来越丰富、技术发展迅猛、产品更新迭代很快，当您阅读本书的时候，可能发现书中的小部分内容与实际不符，欢迎您给我们反馈。如果您有任何宝贵意见，欢迎您发邮件至 [feedback@oe.21vianet.com](mailto:feedback@oe.21vianet.com)。

最后，再次感谢广大读者对世纪互联运营的 Microsoft Azure 的关注，感谢您选择本书！

世纪互联蓝云 技术支持中心 孔德路



# 目 录

## Part 1 商务部分

第一章 管理员账户 .....	2
1.1 管理员的分类及区别 .....	2
1.2 创建管理员账号 .....	5
1.3 更改订阅的服务管理员 .....	10
1.4 设置协同管理员 .....	11
1.5 更改管理员账户的个人资料 .....	13
1.6 重置账户的密码 .....	16
第二章 订阅 .....	21
2.1 订阅的概念以及与账户所有者的关系 .....	21
2.2 创建 Microsoft 账户以及订阅 .....	21
2.3 在一个账户所有者下添加以及管理多个订阅 .....	25
2.4 查看订阅余额以及对订阅进行充值缴费 .....	28
2.5 订阅的几种状态 .....	31
第三章 账单 .....	32
3.1 账单的计费周期 .....	32
3.2 如何下载账单 .....	32
第四章 Azure 服务的计费 .....	34
4.1 计费原则 .....	34

4.2 虚拟机的计费 .....	35
4.3 存储服务的计费 .....	39
4.4 网络资源的计费 .....	43

## Part 2 技术部分

第五章 计算与云服务 .....	50
------------------	----

5.1 虚拟机的使用简介 .....	50
5.2 磁盘和映像的使用 .....	57
5.3 Linux 虚拟机图形化配置 .....	67
5.4 虚拟机扩展的介绍和使用 .....	80
5.5 多网卡虚拟机的配置和使用 .....	87
5.6 云服务的配置和使用 .....	93

第六章 存储 .....	104
--------------	-----

6.1 文件存储常见问题 .....	104
6.2 Azure 文件存储问题疑难解答 .....	106
6.3 普通存储 .....	112
6.4 高级存储 .....	118
6.5 存储管理工具 .....	122

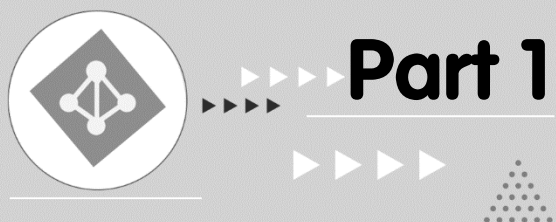
第七章 网络 .....	130
--------------	-----

7.1 IP 地址相关 .....	130
7.2 虚拟网络相关功能 .....	136
7.3 点到站点 VPN .....	142
7.4 站点到站点 VPN .....	155
7.5 虚拟网络网关 .....	171
7.6 Express Route .....	176
7.7 Express Route 混合 VPN .....	179
7.8 BGP VPN 介绍和使用 .....	180

<b>第八章 安全配置</b>	<b>186</b>
8.1 访问控制列表 (ACL)	186
8.2 网络安全组 (NSG)	188
8.3 基于角色的访问控制 (RBAC)	193
8.4 Microsoft Antimalware	197
<b>第九章 负载均衡与高可用设计</b>	<b>201</b>
9.1 面向 Internet 的负载均衡	201
9.2 什么是 Azure Load Balancer	207
9.3 应用程序网关	213
9.4 可用性集	230
9.5 Autoscale\VMSS	234
<b>第十章 备份</b>	<b>244</b>
10.1 Azure 备份功能概述	244
10.2 备份 Azure 虚拟机	245
<b>第十一章 自动化运维</b>	<b>255</b>
11.1 Azure Powershell 的安装与使用	255
11.2 跨平台命令行的安装和使用	259
11.3 Azure Automation 的配置和使用	265
11.4 Azure 资源管理器模板的使用	273
<b>第十二章 内容分发网络</b>	<b>284</b>
12.1 HTTP 加速服务	284
12.2 HTTPS 加速服务	290
12.3 缓存规则	291
12.4 日志查看	294
12.5 FAQ	296

第十三章	Azure 活动目录 .....	304
13.1	Azure 活动目录简介 .....	304
13.2	关于 Azure AD 相关案例分析 .....	305
第十四章	资源组与资源管理器 .....	314
14.1	资源管理器模式 .....	314
14.2	从经典模式迁移到资源管理器模式 .....	316
14.3	资源管理器模式的各类资源 .....	320
14.4	通过 Azure 门户预览创建 Express Route .....	334
14.5	如何在 ARM 模式下去部署 ILB 环境 .....	338
14.6	两台 ARM 虚拟机的负载均衡配置 .....	344
14.7	应用程序网关 .....	357
14.8	使用 PowerShell 来备份 ARM 虚拟机 .....	369
14.9	面向 Internet 的负载均衡 .....	377
第十五章	排错工具与方法 .....	398
15.1	连通性测试 .....	398
15.2	路由检测 .....	401
15.3	抓包工具 .....	402





# 商 务 部 分

---

# 第一章 管理员账户

## 1.1 管理员的分类及区别

管理员是管理 Azure 服务的重要角色，按照各自承担角色的不同，当前分为账户管理员、服务管理员以及协同管理员，每种管理员各司其职，共同努力来将订阅管理得井然有序。

### 1.1.1 账户管理员

首先，账户管理员为最初注册账户时生成的账户，默认形式为 `XX@XX.partner.onmschina.cn`，可以有以下权限。

#### 1. 查看订阅的剩余信用额度以及到期日

首次及随后的每次付款金额应至少为人民币 1000 元。Azure 服务使用额度有效期为 12 个月。当您订阅账户的剩余使用额度为 0，或者信用额度过期后，您的订阅将被停用。当您的订阅处在停用状态，将无法备份数据；激活已停用的订阅后，相关服务需要重新配置。订阅暂停 90 天后，数据将永久删除。

您可以购买额外的 Azure 服务使用额度，额外使用额度从购买日起 12 个月有效。

这就意味着您订阅的剩余信用额度和到期日尤为重要。

第一种方法是账户管理员可以登录 `account.windowsazure.cn` 来直接进行查看，如图 1.1-1 所示。

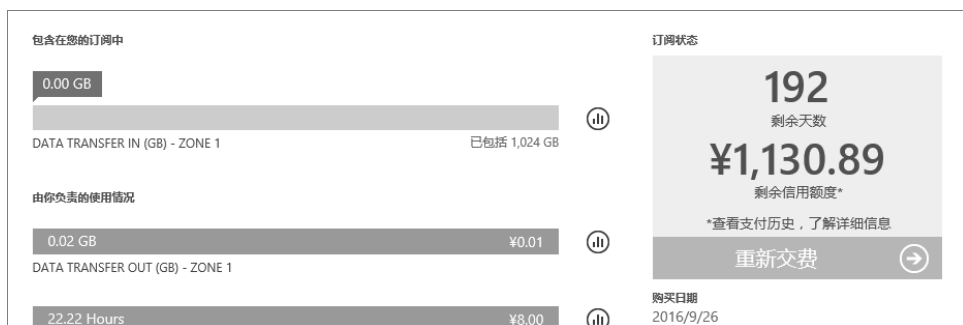


图 1.1-1

第二种方法是：在您注册之初，您需要输入您的邮箱，而您的邮箱则作为我们联系您的重要联系方式。每周一系统会自动发送余额通知邮件，通知邮件会包含剩余信用额度以及剩余天数。

为保证您能正常接收系统寄发的 Azure 剩余余额通知邮件，需要确保您填写的邮箱的有效性以及正确性。如需修改，请按照以下步骤操作。

(1) 登录 <https://account.windowsazure.cn/profile>【个人资料/Profile】，如图 1.1-2 所示。

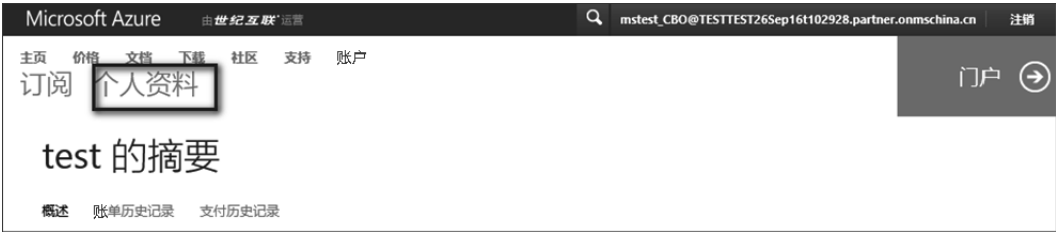


图 1.1-2

(2) 单击【编辑详细信息/Edit Details】，如图 1.1-3 所示。

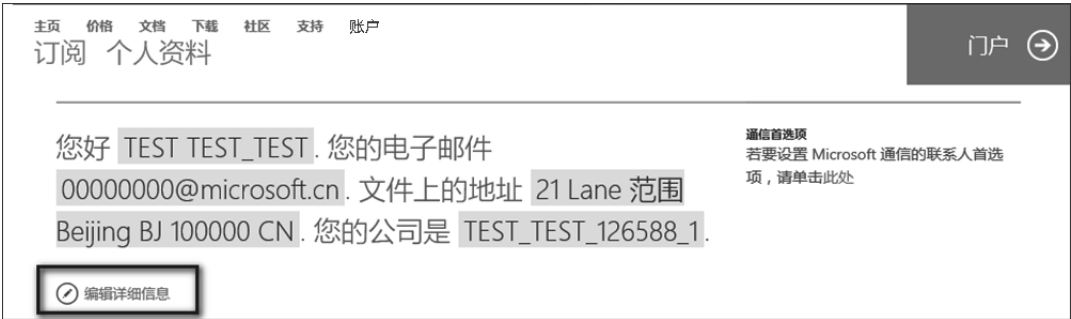


图 1.1-3

(3) 修改【联系人电子邮件/Contact Email】，单击“√”确定，如图 1.1-4 所示。

图 1.1-4

2. 设置服务管理员

在账号创建最初，服务管理员会默认与账号管理员一致，如果需要将职能分派给其他人，账户管理员则有权对服务管理员做出修改（详情见下一节），如图 1.1-5 和图 1.1-6 所示。



图 1.1-5



图 1.1-6

3. 创建新订阅

默认情况下，每个账户下会有一个订阅，而在当前实名认证流程完善的基础上，如果标准预付费账户有多个部门使用 Azure 服务并且希望账单可以分开管理，则有必要在现有一个订阅的基础上，添加多个订阅进行管理。

1.1.2 服务管理员

访问并管理开发人员门户上的订阅和开发项目

作为管理 Azure 部署服务的最高管理员，在账号创建最初，服务管理员会默认与账号管理员一致，即可以登录经典管理门户（manage.windowsazure.cn）以及新门户（portal.azure.cn）管理部署。但此服务管理员可以修改，下一节会有详细表述。

1.1.3 协同管理员

如果服务管理员一人管理订阅较为困难，则可以添加其他人员来协助共同管理订阅服务，顾名思义，此管理员则称之为协同管理员。

服务管理员负责管理订阅下的协同管理员，控制对订阅中服务的管理权限。

而协同管理员与服务管理员对订阅下的服务权限并无不同，协助服务管理员对整个订阅下的服务有管理权限，如果需要限制某个账号对某个服务的权限，可以使用新 portal（portal.azure.cn）来进行设置。

## 1.2 创建管理员账号

上一节所讲到的三种管理员是针对订阅的管理权限的不同所划分出来的。如果要设置以上权限，首先则需要先将账号创建出来，然后才能分配权限。

创建账号则属于域名下的用户管理，通常同域名下的管理分为两种角色，分别是全局管理员和普通用户。

当注册 Azure 账户时，会获得一个格式为 `XX@XXX.partner.onmschina.cn` 的账号，而此时，就自动成为管理域名 `XXX.partner.onmschina.cn` 的全局管理员，拥有最高管理员权限，可以创建同一域名下的新用户，并且在创建时，就可以设置这些新用户的权限为全局管理员或普通用户。

全局管理员可以创建新用户，管理现有的活动用户：重置密码，更改重置密码邮箱以及删除现有活动用户，如图 1.2-1 所示。

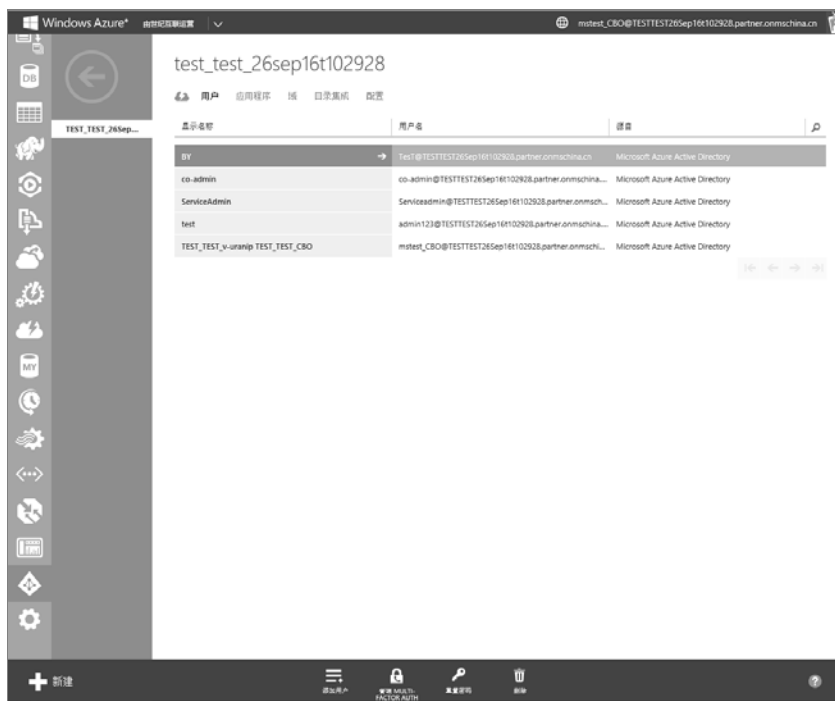


图 1.2-1

普通用户并没有查看 AD 以及管理 AD 的权限，所以也无法创建以及查看、管理现有目录下的用户。

新账号的创建如下所示。

第一种情况：如果服务管理员是全局管理员，可以直接在 `manage.windowsazure.cn` Active Directory 里添加活动用户。

以下步骤为在 AD 里添加活动用户。

(1) 单击 AD 下的目录，如图 1.2-2 所示。

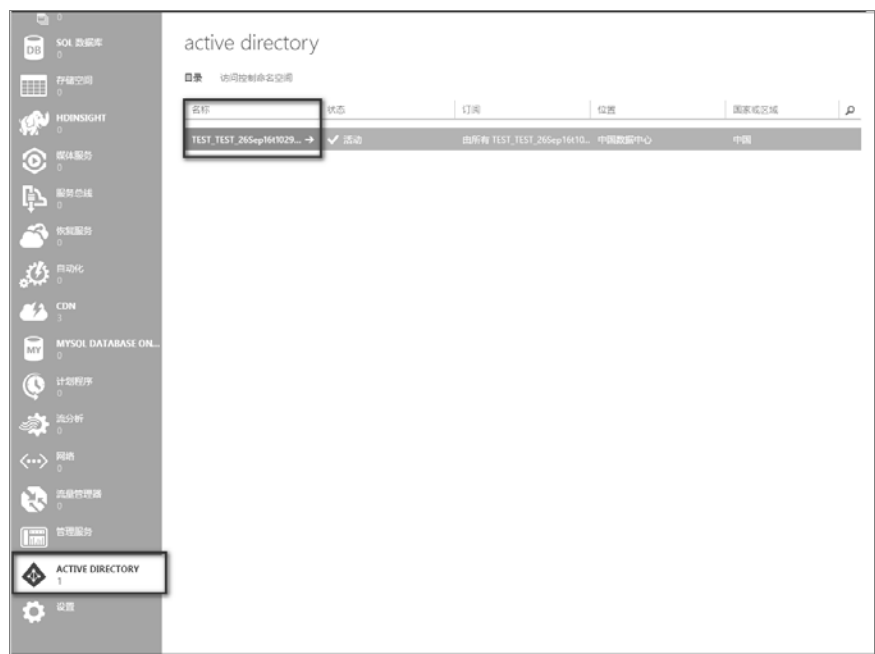


图 1.2-2

(2) 单击“用户”，如图 1.2-3 所示。



图 1.2-3

(3) 现在可查看现有目录下的所有活动用户。如果需要添加，可单击页面最下方的“添加用户”，如图 1.2-4 所示。



图 1.2-4

(4) 在第一个页面填写用户名，如图 1.2-5 所示。



添加用户

告诉我们有关此用户的信息

用户类型

你的组织中的新用户

用户名

test

TESTTEST26Sep16t102928.partner.on

2 3

图 1.2-5

(5) 添加名字、姓氏、显示名称。

角色请按照需求选择用户或者全局管理员，之后单击“下一步”。

用户和全局管理员的权限区别在上一节有所说明：用户没有权限来查看和管理 AD；而全局管理员作为目录（域名）下的管理员，有权限来查看和管理编辑同一域名下的所有用户，如图 1.2-6 所示。



添加用户

用户配置文件

名字

admin

姓氏

test

显示名称

test

角色

用户

多重身份验证

☐ 启用多重身份验证

3

图 1.2-6

（6）单击“创建”，如图 1.2-7 所示。



图 1.2-7

（7）最后一个页面已成功创建一新用户，新密码可以通过邮件发送给相关用户，单击“√”确认，如图 1.2-8 所示。



图 1.2-8

以上的情况为服务管理员为全局管理员的情况，可以直接在 AD 里管理活动用户。

第二种情况：如果服务管理员并非全局管理员，则需要域名下的全局管理员（账户管理员默认为全局管理员）在 portal <https://portal.partner.microsoftonline.cn> 里进行添加。

（1）单击“管理”—“活动用户”，单击“+”添加活动用户，如图 1.2-9～图 1.2-11



所示。

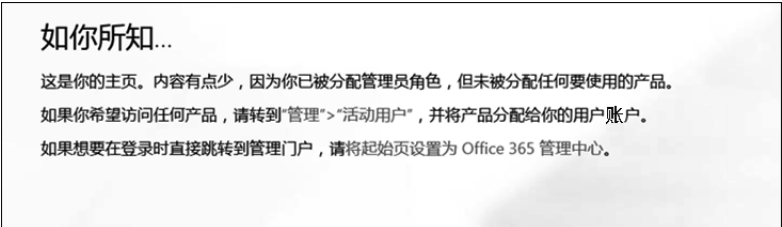


图 1.2-9



图 1.2-10

### 创建新用户账户

姓氏

名字

XX

XX

\* 显示名称

XXXX

\* 用户名

XX

@

testtestccname02.p

自动生成的密码 | 键入密码

新密码将显示在下一页中

☒ 让用户在下次登录时更改密码。

\* 通过电子邮件将密码发送给以下收件人

XX@163.com

为此用户选择许可证:

目前没有要分配的许可证。购买更多许可证

创建

取消

图 1.2-11

(2) 出现以下页面时，代表已成功创建了活动用户，如图 1.2-12 所示。



图 1.2-12

活动用户在 AD 里或者在 portal <https://portal.partner.microsoftonline.cn> 添加之后，如果需要给账号分配权限，则需要查看下一节。

### 1.3 更改订阅的服务管理员

默认情况下，服务管理员和账户管理员一致，但账户管理员可以有权限来更改订阅的服务管理员。服务管理员的账号需要是和账户管理员同域名并且是同域名下的活动用户，是否为活动用户则需要账户管理员登录 <https://portal.partner.microsoftonline.cn> 进行查看。如果并不在活动用户中，则按照上一节所述进行添加，如已存在并需要修改订阅的服务管理员，请看以下步骤。

(1) 使用账户管理员登录 [account.windowsazure.cn](https://account.windowsazure.cn)，单击“订阅”，如图 1.3-1 所示。

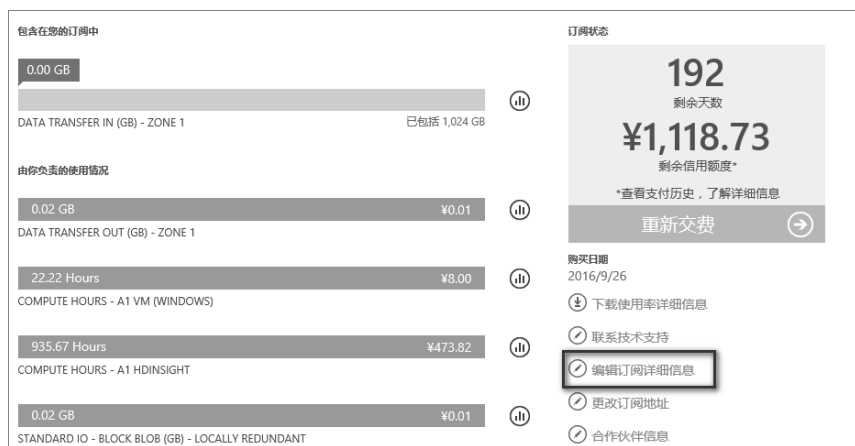


图 1.3-1

(2) 在弹出的对话框中可编辑订阅名称以及服务管理员，单击“√”进行确认，如图 1.3-2 所示。

编辑完服务管理员，则可使用新更改的服务管理员来登录 [manage.windowsazure.cn](https://manage.windowsazure.cn) 和 [portal.azure.cn](https://portal.azure.cn)，并且通过“设置”—“管理员”来查看现有的服务管理员。



图 1.3-2

## 1.4 设置协同管理员

### 1.4.1 添加协同管理员

首先，协同管理员的账号需要是和服务管理员同域名并且是同域名下的活动用户。如果服务管理员为域名下的全局管理员，则可直接在 Azure 中的 Active Directory 进行查看；如果服务管理员并非全局管理员，则需要账户管理员登录 <https://portal.partner.microsoftonline.cn> 进行查看。

如果需要添加的协同管理员并不存在于域名下的活动用户中，则需要按照第二节所述的创建账号步骤进行添加。如果已经存在，则按照以下步骤分配权限。

(1) 使用服务管理员登录 [manage.windowsazure.cn](https://manage.windowsazure.cn)，单击“设置”—“管理员”，单击页面下侧的“添加”，如图 1.4-1 所示。



图 1.4-1

(2) 在电子邮件地址填写刚刚添加的活动用户账号，勾选需要添加的订阅 ID，单击“√”确认，如图 1.4-2 所示。



图 1.4-2

(3) 之后查看管理员界面，则可看到服务管理员以及协同管理员，如图 1.4-3 所示。



图 1.4-3

需要注意的是，添加协同管理员，只能添加与服务管理员同一域名下的活动用户，无法添加不同域名下的活动用户。

### 1.4.2 删除协同管理员

选中需要删除的管理员，单击页面下侧即可删除，如图 1.4-4 所示。



图 1.4-4

## 1.5 更改管理员账户的个人资料

### 1.5.1 更改账户管理员资料

(1) 账户管理员应登录 <https://account.windowsazure.cn/profile> 【个人资料/Profile】，如图 1.5-1 所示。



图 1.5-1

(2) 单击【编辑详细信息/Edit Details】，如图 1.5-2 所示。



图 1.5-2

(3) 修改以下个人信息，包括姓名，联系人电子邮件，电话号码以及相关的地址信息，单击“√”确定，如图 1.5-3 所示。

名字

test\_test\_fff

联系人电子邮件

test\_test\_ccname02@o

姓氏

test\_test\_lll

联系人电话号码

111111111

地址行 1

beijing456

组织

test\_test\_fccname02\_uj

地址行 2

市/县

beijing

区

省/直辖市/自治区

北京

邮政编码

100100

图 1.5-3

1.5.2 更改其他管理员资料

全局管理员应登录 <https://portal.partner.microsoftonline.cn>，勾选需要修改的用户，单击“编辑”，如图 1.5-4 所示。

单一登录: 设置 | 了解详细信息  
Active Directory 同步: 设置 | 了解详细信息  
更改用户的密码过期策略: 立即更改

选择视图: 所有用户

+

✖

🔍

🔄

<input type="checkbox"/>	显示名称	用户名	状态
<input type="checkbox"/>	Jean	Jean@testtestccname02.partner.onmsch...	在云中
<input type="checkbox"/>	test_test_fccname02 test_test_jccname02	testtestccname02@testtestccname02.pa...	在云中
<input checked="" type="checkbox"/>	XXXX	XX@testtestccname02.partner.onmschin...	在云中

👤

XXXX

🔑 重置密码

✏️ 编辑用户角色

🗑️ 删除

✏️ 编辑

已分配的许可证

无许可证 编辑

图 1.5-4

在第一个页面，可以修改姓名，如图 1.5-5 所示。

• 14 •

XXXX

详细信息

角色

设置

许可证

更多

名称

姓氏

XX

名字

XX

\* 显示名称

XXXX

\* 用户名

XX

@ testtestccname02.partner.onmschina.cn

其他详细信息

图 1.5-5

角色页面可更改角色设置、用户和全局管理员设置，全局管理员若需要修改重置密码的发送邮件地址即备用电子邮件地址，则在此处来修改，如图 1.5-6 所示。

←

XXXX

详细信息

角色

设置

许可证

更多

分配角色

选择您希望分配给此用户的管理员角色并保存更改 了解有关管理员角色的更多信息

☐ 无管理员访问权限

☒ 全局管理员  
对所有管理功能具有访问权限

☐ 受限管理员访问权限  
对有限的管理功能具有访问权限

\* 备用电子邮件地址

我们将使用此电子邮件地址作为备用电子邮件地址，以帮助用户重置密码。 了解有关恢复丢失的密码的更多信息

图 1.5-6

另外可以更改其设置，完成之后单击“保存”即可，如图 1.5-7 所示。

←

XXXX

详细信息

角色

设置

许可证

更多

设置登录状态

☒ 已允许  
用户可以登录和访问服务。

☐ 已阻止  
用户无法登录或访问服务。

图 1.5-7

### 1.5.3 全局管理员更改本人信息以及备用邮箱地址

单击账户右上角，可查看账户，单击“个人信息”，可编辑电话或者备用电子邮件地址，如图 1.5-8 和图 1.5-9 所示。



图 1.5-8



图 1.5-9

## 1.6 重置账户的密码

如果您记得账户，但是忘记密码，一般有以下几种方式重置密码。

一种是自行重置密码。

点开链接 <https://manage.windowsazure.cn> 或者 <https://account.windowsazure.cn>，输入账户，单击“无法访问您的账户”，如图 1.6-1 所示。

单击“工作或学校账户”，如图 1.6-2 所示。





图 1.6-1



图 1.6-2

输入出现的验证码，单击“下一步”，如图 1.6-3 所示。

单击“电子邮件”，备选电子邮件地址将会收到一封包含验证码的邮件，如图 1.6-4 所示。



图 1.6-3



图 1.6-4

之后保持此页面不关闭，方框里填写您收到的代码，单击“下一步”，如图 1.6-5 和图 1.6-6 所示。



图 1.6-5



图 1.6-6

然后输入新密码以及确认密码，单击“完成”即可，如图 1.6-7 所示。



图 1.6-7

需要注意的是，如果账户是域名下的普通用户，则没有权限自主来重置密码，自主重置时会出现以下提醒，如图 1.6-8 所示。

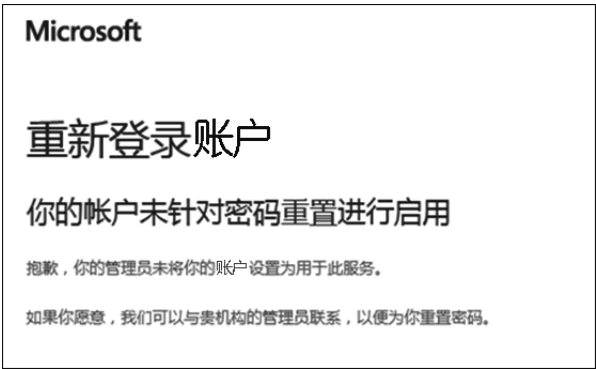


图 1.6-8

所以域名下的普通用户如果需要重置则需要按照第二种方法来进行。  
第二种方法，请您账户域名的全局管理员登录 office 365 来帮忙重置密码。  
使用全局管理员登录 <https://portal.partner.microsoftonline.cn>，单击“管理” — “活动用户”，则可查看现有同域名下的所有活动用户，如图 1.6-9 所示。



图 1.6-9

选中要重置密码的用户，单击“重置密码”，如图 1.6-10 所示。



图 1.6-10

修改通过电子邮件发送结果，可以选择让此用户在下次登录时更改密码。

单击“重置”后会出现新的密码，将新密码发送给需要修改密码的用户即可，如图 1.6-11 所示。

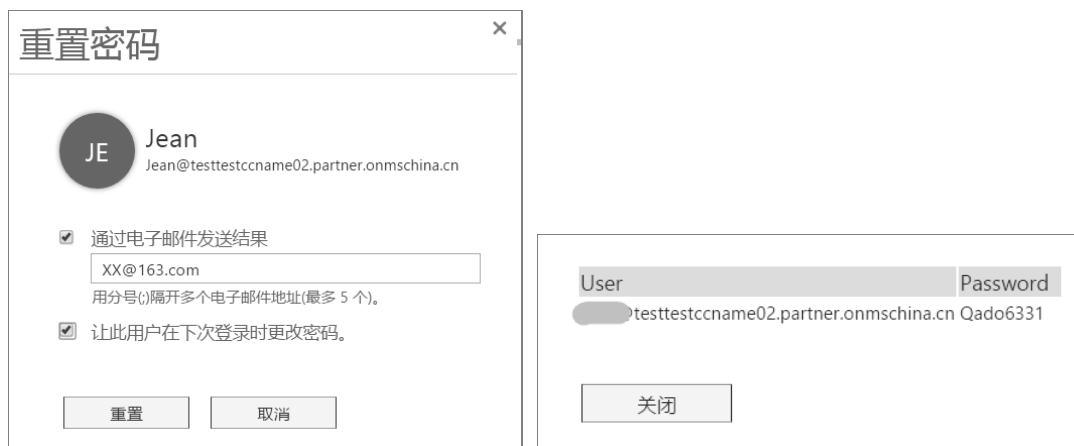


图 1.6-11

如果您无法通过自主重置密码或者请其他管理员帮您重置密码，请联系 21V support 团队。

## 第二章 订 阅

### 2.1 订阅的概念以及与账户所有者的关系

订阅是提前预定的一种方式，它为用户使用 Microsoft Azure 的所有资源、支付，以及账单提供了链接入口。用户需要对订阅进行充值的操作才能开始使用 Azure 的服务。

在 Microsoft Azure 中，订阅与账户是绑定的，一个账户中可以购买多个订阅，但每个订阅的费用以及计费和使用是独立分开的，没有关联。但对于订阅来说，仅只能与一个账户做绑定。

用户可以用订阅的名称进行修改的操作，以便区分不同的订阅，例如：用户可以将订阅命名为财务部、市场部、营销部，以对多个部门分开做管理。

账户所有者可以对其下的订阅进行管理，例如：下载账单、查看余额及有效期、充值缴费、编辑订阅的名称，以及更改服务管理员等。

### 2.2 创建 Microsoft 账户以及订阅

注册 PIA 账户直到生成订阅，大概分为以下几步。

填写身份验证信息—获得账号和密码—填充支付信息—在支付平台上完成支付。

(1) 登录 Azure 官网 [www.azure.cn](http://www.azure.cn)，单击右上角的“我要购买”，如图 2.2-1 所示。

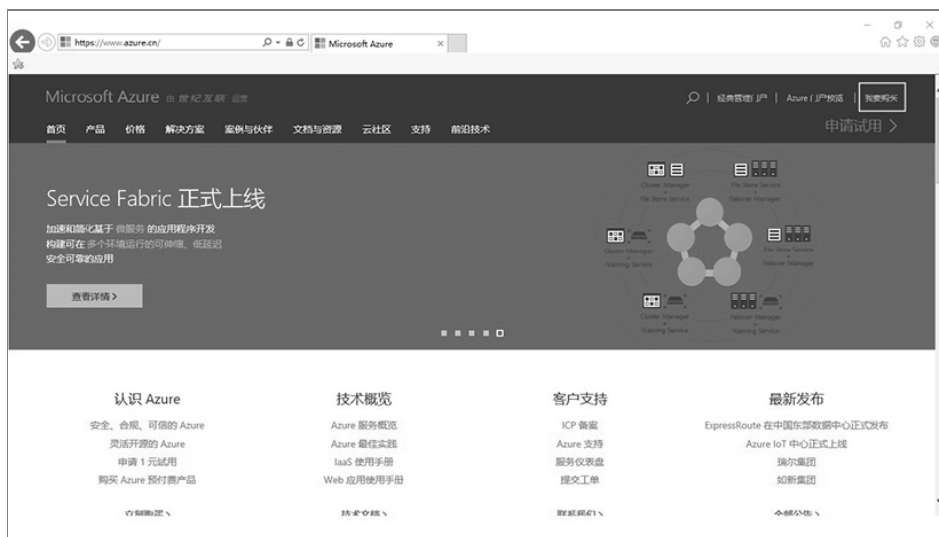


图 2.2-1

(2) 页面跳转到“Azure 预付费产品购买申请表”页面，进行简单的手机验证即可进入下一步，如图 2.2-2 所示。

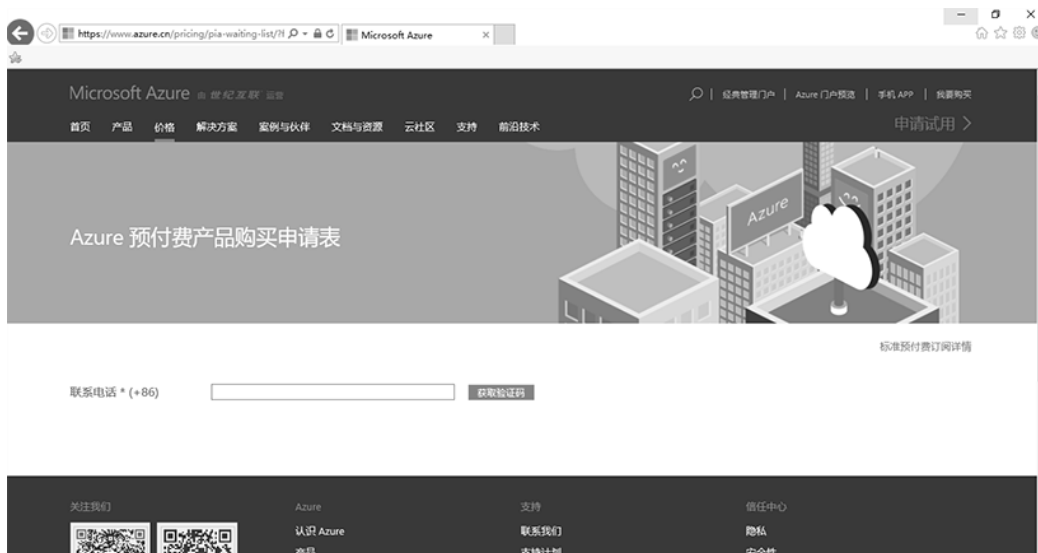


图 2.2-2

(3) 然后来到身份查检页面，可以根据需要选择“个人申请”或“企业申请”，如图 2.2-3 所示。



图 2.2-3

(4) 快速填写好信息后就可以提交了，如图 2.2-4 和图 2.2-5 所示。

(5) 申请资料审核通过后进入注册页面，填写信息创建域名和密码，完成手机验证，然后去付款，如图 2.2-6 所示。

选择申请类型：☒ 个人申请 ☐ 企业申请

为了遵守相关法律法规规定，保护您的账户安全，您需要填写真实的身份信息。我们会严格遵守 [隐私声明](#) 条款妥善保管您所提交的信息。

您的姓名\*

您的邮箱\*

身份证号码\*

身份证正面扫描件\* + 上传附件

身份证反面扫描件\* + 上传附件

如果您已经申请过试用，欢迎购买标准预付费订阅，如要购买请点击 [这里](#)。

为了遵守相关法律法规规定，保护您的账户安全，您需要填写真实的身份信息。我们会严格遵守 [隐私声明](#) 条款妥善保管您所提交的信息。

图 2.2-4

选择申请类型：☐ 个人申请 ☒ 企业申请

为了遵守相关法律法规规定，保护您的账户安全，您需要填写真实的身份信息。我们会严格遵守 [隐私声明](#) 条款妥善保管您所提交的信息。

联系人姓名\*

联系人邮箱\*

联系人身份证号码\*

联系人身份证正面扫描件\* + 上传附件

联系人身份证反面扫描件\* + 上传附件

企业名称\*

组织机构代码\*

营业执照扫描件\* + 上传附件

图 2.2-5

Microsoft Azure 由世纪互联运营

Microsoft Azure 欢迎

### 关于您

名字  姓氏

联系人电子邮件地址  组织名称(可选)

### 您的登录信息

域名  .partner.onmschina.cn

新建用户 ID

@azuretest001.partner.onmschina.cn

创建新密码  确认新密码

### 您的手机号码和验证

☒ 发送短信

中国 (+86)

358475

您输入的验证代码不正确。单击[此处](#)访问最近的支持中心。

中文(简体) 隐私与 Cookie 合法 支持 向我们提供反馈 Microsoft Azure 由世纪互联运营

图 2.2-6

完成该步之后，即可获得账号和密码信息。

(6) 登录账户，输入刚才注册设定的密码即可，如图 2.2-7 所示。

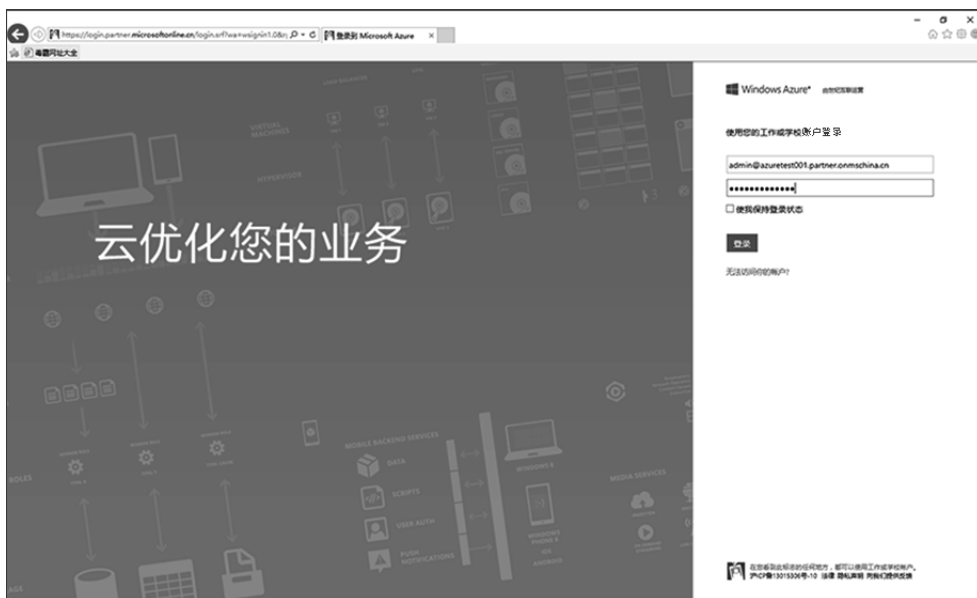


图 2.2-7

(7) 可以选择“支付宝”、“银联在线支付”、“电汇”三种付款方式，支付宝和银联在线支付最低充值金额是 1000 元，电汇即线下付款最低支付金额是 5000 元，如图 2.2-8 所示。

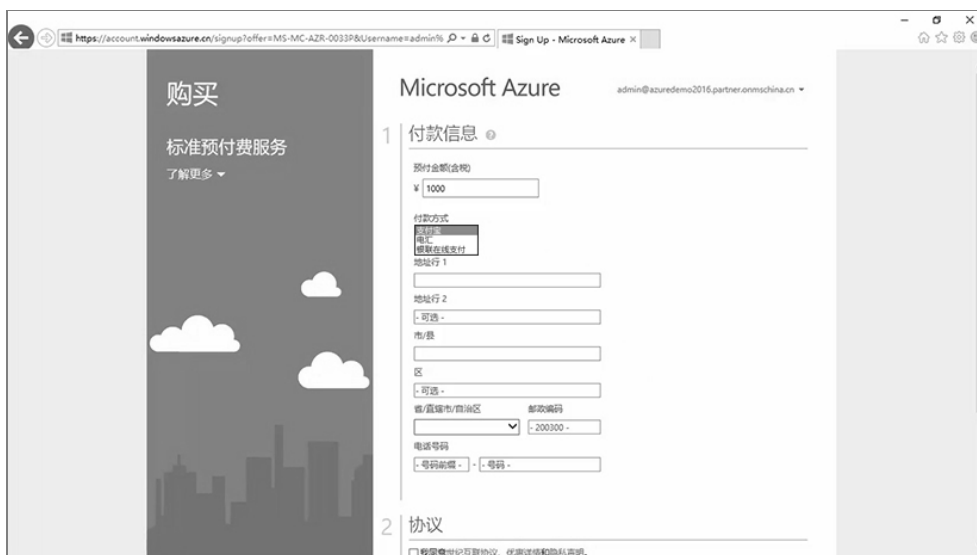


图 2.2-8

(8) 选择支付方式，在支付平台上完成支付后，系统自动带您来到账户管理界面，此



时就可以看到生成了一个订阅，并且可以看到账号的信用额度信息，这时您可以来到管理门户使用相关服务了。图 2.2-9 和图 2.2-10 是使用支付宝支付 1000 元的截图。

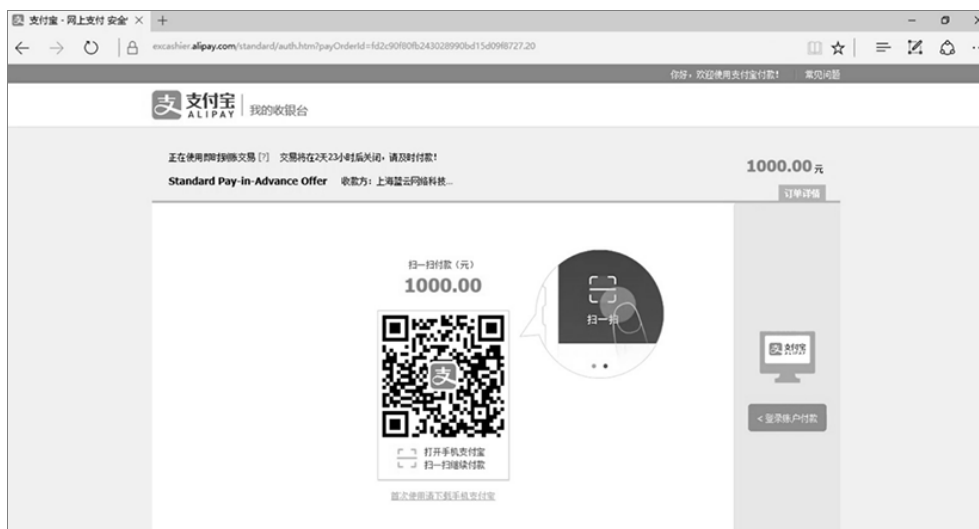


图 2.2-9

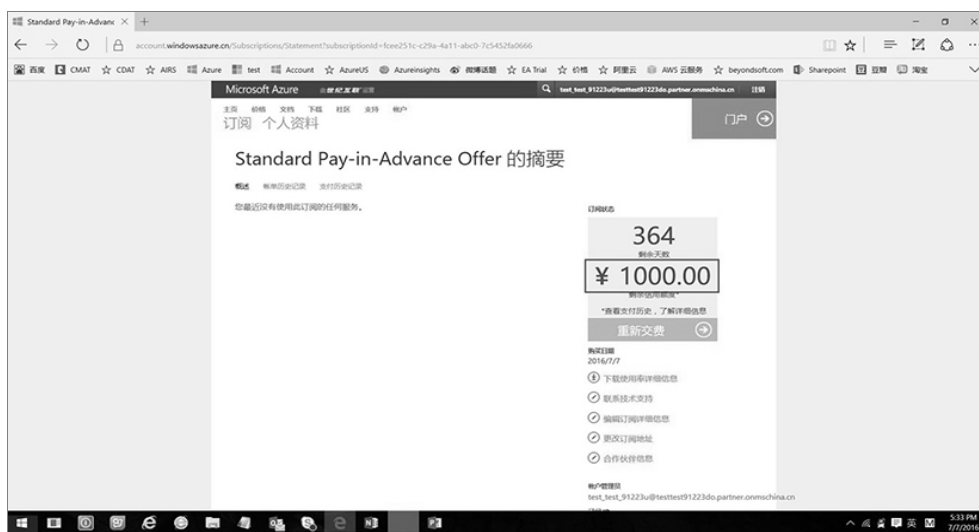


图 2.2-10

## 2.3 在一个账户所有者下添加以及管理多个订阅

### 1. 添加多个订阅

在一个 PIA 账号下添加多个订阅的步骤如下。

- (1) 单击链接 [account.windowsazure.cn](https://account.windowsazure.cn)，单击“账户中心”，使用账户和密码登录，

如图 2.3-1 所示。



图 2.3-1

(2) 保持页面不关闭，然后在同一个浏览器打开另外一个新的页面，输入地址 [azure.cn](https://azure.cn)，单击“我要购买”，如图 2.3-2 所示。



图 2.3-2

(3) 输入之前申请账号时使用的手机号，单击“获取验证码”，如图 2.3-3 所示。

## Azure 预付费产品购买申请表

下列选项均为必填项

联系电话

+86 |

获取验证码

[标准预付费订阅详情 >](#)

图 2.3-3

(4) 收到验证码之后，单击“验证”，之前是个人申请就选择“个人申请”，之前是企业申请就选择“企业申请”，单击“提交”，如图 2.3-4 所示。

(5) 提交之后直接会跳转到您的账户的付款页面，在这里可以更改支付方式。勾选“我同意……”，单击“购买”，如图 2.3-5 所示。



图 2.3-4



图 2.3-5

付款成功后会生成一个新的订阅。

2. 如何管理一个账号下的多个订阅

可以创建多个协同管理员，通过给不同的协同管理员分配不同的订阅，实现分工管理的需求。

标准预付费订阅生成之后，默认的名称都是“标准预付费服务”，可以按照如下方法修改订阅名称以便区分多个订阅。

(1) 选择“订阅”，单击“订阅名称”，出现如图 2.3-6 所示的界面，单击右侧的“编辑订阅详细信息”。



图 2.3-6

(2) 在订阅名称处可以修改订阅名称，最后单击右下角的“√”保存，如图 2.3-7 所示。



图 2.3-7

## 2.4 查看订阅余额以及对订阅进行充值缴费

### 1. 查看订阅余额

登录 account portal (<https://account.windowsazure.cn>)，选择“订阅”，单击“订阅名称”，在出现的界面右侧可以查看订阅剩余金额，如图 2.4-1 所示。



图 2.4-1

另外，您预留的联系人邮箱将每周收到剩余金额及有效期的提醒邮件。您也可以按照以上方法随时登录账户管理界面查看剩余金额及有效期，建议您根据实际使用量进行评估，

预留充足的余额。

## 2. 对订阅进行充值交费

(1) 登录账户管理门户 account portal (<https://account.windowsazure.cn>)，选择“订阅”，单击“要充值的订阅名称”，在出现的界面右侧单击“重新交费”，如图 2.4-2 所示。



图 2.4-2

(2) 目前有三种支付方式可以选择：支付宝、电汇、银联在线支付。若选择“支付宝”或“银联在线支付”线上汇款方式，在单击“重新充值”后，按照提示输入充值的金额（最低充值金额为 1 000 元，最高不可超过 150 000 元），单击“√”，如图 2.4-3 所示。若选择电汇方式，最低充值金额为 5 000 元，其操作方式，可参考官网相关“电汇”说明 (<https://www.azure.cn/pricing/billing/azure-wire-transfer-overview>)。



图 2.4-3

(3) 这里我们以选择“支付宝”支付为例，单击“下一步”，出现如图 2.4-4 所示的界面，填写充值金额。



图 2.4-4

(4) 单击右下角的“转至支付宝”，进入支付宝付款平台，如图 2.4-5 所示。



图 2.4-5

(5) 支付成功后，您可以返回如下界面查看剩余金额，以确定是否充值成功，如图 2.4-6 所示。

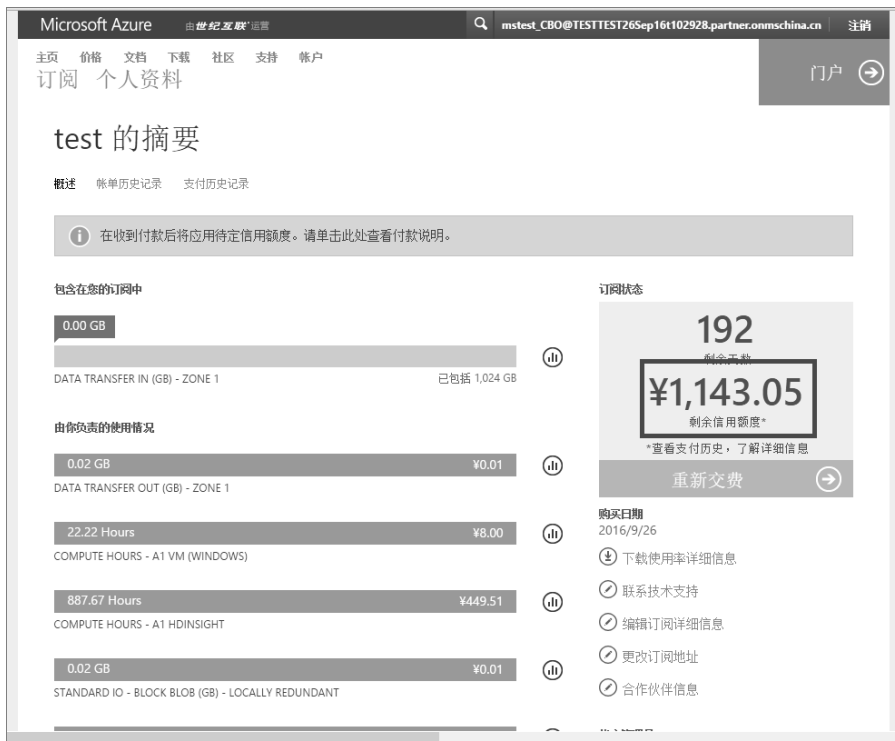


图 2.4-6

需要注意的是，通过银联在线支付，在订单提交后，需在 15 分钟内完成支付；如果超过 15 分钟，可能导致付款不成功，需要再次重新提交订单，通过支付宝支付，订单提交后有效期是 3 天。

## 2.5 订阅的几种状态

订阅一般有以下几种状态：挂起、有效、已取消。

**挂起：**当完成填写身份验证信息、获得账号和密码、填充支付信息这几步，但未在支付平台支付成功时，订阅在 account portal 中的状态将是“挂起”，此时的订阅还未开通，不能使用。

**有效：**客户进入支付平台支付成功后，订阅在 account portal 中的状态将是“有效”，只有处于“有效”状态的订阅才是可以使用的。

当订阅超过有效期或余额用完后，订阅将会自动停止，但是在 account portal 仍会显示为“有效”的状态，但是此时的订阅已经不可以使用了。

**已取消：**如果订阅被停的 90 天内客户没有充值，订阅里的数据将会自动删除，订阅在 account portal 将会显示为“已取消”状态。

另外，当订阅由于违规或其他原因被后台手动停止后，也将显示为“已取消”状态。

## 第三章 账 单

对于每一个 Azure 用户，实时了解自己的账户使用情况，以及及时充值或调整服务配置避免额外的费用是非常必要的。下面，我们就介绍一下账单的相关内容。

### 3.1 账单的计费周期

由于 Azure 不采用自然月作为出账周期，所以产生了计费周期的概念。在此要先解释一下账单日的概念，账单日是指用户进行首次充值操作时的日期，账单月即计费周期就是从账单日起之后的一整个月。

例如，某一用户在 2016 年 9 月 26 日注册了 Azure 账户，并充值购买了一个在线标准预付费订阅，则此用户的账单日为每月的 26 日，计费周期为当月 26 日至次月 25 日，如图 3.1-1 所示。

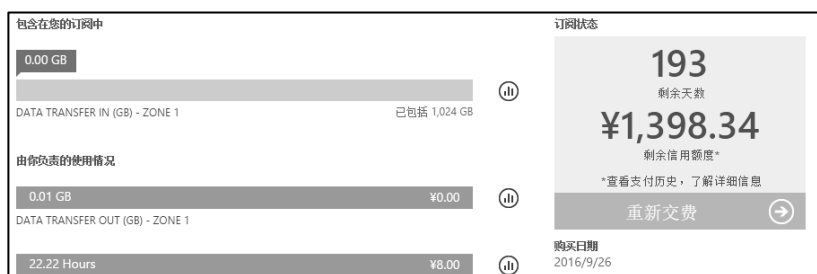


图 3.1-1

注：本书所述均为一般场景，如遇特殊情况，可通过以下链接联系相关支持：<https://support.windowsazure.cn/support/support-azure/>。

### 3.2 如何下载账单

如前文中提到，Azure 针对不同的角色，是通过不同的登录网址来约束其权限的。Azure billing 门户 (<https://account.windowsazure.cn>) 是专门用来管理订阅信息和查看账单的门户，只有账户所有者才能登录，服务管理员和协同管理员都没有权限进入此门户。

账户所有者下载账单的操作步骤如下。

(1) 登录 Azure billing 门户 (<https://account.windowsazure.cn>)。

用户在输入 Azure billing 门户网址之后，可以选择图 3.2-1 右上角的“登录”按钮，输入账户所有者账户和密码登录，然后选择“账户中心”。





图 3.2-1

进入 Azure billing 门户订阅主界面之后，默认账户下订阅是收起状态，只显示订阅名称和订阅当前的状态。

(2) 单击“订阅”，选择“下载使用率详细信息”，如图 3.2-2 所示。



图 3.2-2

单击“下载使用率详细信息”之后会跳转到**账单历史记录**界面，在此界面用户每个计费周期的账单会按时间顺序显示。

(3) 选择要下载的计费周期的账单，单击相应周期的“下载使用量”按钮，就可以得到 Excel，即用户的账单，如图 3.2-3 所示。

test 的摘要		
概述 账单历史记录 支付历史记录		
下方显示的金额不包括调整金额，如折扣、信用额度 and 退款。		
当前期间	查看当前声明	下载使用量
2017/2/2 - 2017/3/1	下载使用量	¥560.13
2017/1/2 - 2017/2/1	下载使用量	¥254.28
2016/12/2 - 2017/1/1	下载使用量	¥56.33

图 3.2-3

## 第四章 Azure 服务的计费

### 4.1 计费原则

Azure 提供了多达几十种服务，用户可以灵活地选择其中一个或多个服务，并且只需要为其使用的服务付费。对于用户未使用的服务，Azure 不会向用户收取费用。每项 Azure 服务的计费标准也不一样，但它们都具有以下一些共同的特征。

#### 1. 按实际使用量计费

如同移动通信运营商对用户使用的移动语音服务按通话时长计费，对 4G 互联网接入服务按流量计费一样，Azure 的各项服务也是按用户的实际使用量计费。计量的单位则有所不同，对于计算类的服务，计算的是使用时长，单位为分钟或小时；而对于数据存储与传输类的服务，则以 GB 为单位计量。

#### 2. 阶梯定价

有些 Azure 服务，采取“阶梯定价”的方式计费，表 4.1-1 是某项 Azure 存储服务的计费标准。

表 4.1-1

每月存储容量 (TB)	单价 (¥/GB)
0~1	0.88
1~50	0.7334
50~500	0.66
500~1000	0.5866
1000~5000	0.55

从表 4.1-1 可以看到，当用户使用量大时，其单位价格相对便宜。举个例子，假设某用户使用该存储服务存储 60TB 数据达一个月，则费用如表 4.1-2 所示。

表 4.1-2

每月存储容量 (“梯度”范围) (TB)	单价 (¥/GB)	“梯度”内用量 (GB)	“梯度”内费用 (¥)
0~1	0.88	1×1024	901.12
1~50	0.7334	49×1024	36 799.08
50~500	0.66	10×1024	6 758.40
合计			44 458.60

而不是下面这样计算：

$$0.88 \times 1024 \times 60 = 54067.2$$

因为存储的数据越多，用户为每 GB 支付的费用就越少，用户就可以从规模经济中不断受益。

### 3. 提供多级别的服务

有些 Azure 服务按服务能力划分了多级别的服务标准，来满足用户的不同工作负载的需求。如 Azure SQL Database 服务，被划分为基本级别、标准级别和高级级别，标准级别和高级级别还进一步划分了 S0-S3，P1-P15。

各级别的 SQL Database 服务能够提供从 2GB~1TB 不同的数据库大小，提供从 7 天到 35 天不同的定点备份，当然其价格也随着所提供能力的提升而提高。

### 4. 注意服务间的依赖关系

有些 Azure 服务的提供，是依赖于其他某项服务的，被依赖的服务也会按用户的实际使用量计费。如 Azure 虚拟机服务，依赖于 Azure 页 Blob 存储服务（后者为其提供磁盘存储服务）和数据传输服务（后者为其提供互联网接入服务），相应地，用户也需按这两项服务的实际使用量和对应的价格标准付费。

下面，我们就介绍几个 Azure 服务的计费标准。这些 Azure 服务被用户广泛使用，了解它们的计费标准，将有助于您把成本因素考虑到您的应用规划中，帮助您的企业充分地从事务中受益。注：本书介绍的计费信息，旨在帮助用户理解 Azure 计费逻辑，服务提供商随时有可能调整价格标准和计费策略。

## 4.2 虚拟机的计费

Azure 按使用时长计算虚拟机对计算资源（CPU，内存和磁盘缓存）的使用量，单位是分钟（Azure 网站上所示的价格是按小时汇总的，即假设虚拟机持续开启 1 小时的价格）。不同的虚拟机配置，其单位时间成本也不一样，取决于用户对其软硬件配置的需求。与采购传统的服务器类似，软硬件配置，是否配置服务器集群都是影响采购成本的因素。具体影响 Azure 虚拟机成本的因素如下。

### 1. 计算资源的投入（CPU，内存，磁盘缓存等）

CPU、内存、虚拟内存（对于 Windows 操作系统，虚拟内存是以页面文件的形式提供的；对于 Linux 操作系统，虚拟内存是以 SWAP 分区的形式提供的）是主要的计算资源，这些计算资源分配的高低，决定了虚拟机的处理能力，也决定了虚拟机的成本。

Azure 提供了多个系列，不同配置的虚拟机，适用于各种用户需求。但不支持用户自定义配置。

### 2. SSD 的使用

Solid State Drive，通常被称为“固态硬盘”，读写性能优于传统硬盘，其售价也远高于硬盘。在传统数据中心的 SSD 主要在“冷”、“热”分层的存储结构中用于存储“热”数据，提高“热”数据的命中率。

在 Azure 上，也提供了类似的解决方案：

每台 Azure 虚拟机默认配置了 2 块虚拟磁盘，分别用于操作系统和缓存。对于 Windows

操作系统，其页面文件默认存储在专用于缓存的磁盘上；对于 Linux 操作系统，缓存磁盘被“mount”在 SWAP 分区上。

在 D、F 等系列的虚拟机中，用于缓存的磁盘实际上是 SSD，Windows 页面文件和 Linux SWAP 分区存储在 SSD 上，将显著提升虚拟机的性能。用户也可以配置虚拟机中的应用，将“热”数据存储在 SSD 缓存中，提高“热”数据的命中率。

由于 Azure 缓存盘的配置特点，用户位于缓存盘上的数据不会被持久保存，建议用户为虚拟机附加用于持久保存数据的磁盘。Azure 也提供了 Ds 和 Fs 系列的虚拟机，不仅用于缓存的磁盘采用 SSD，用于存储操作系统和用户数据的磁盘也可以全部或部分采用 SSD，满足对磁盘 I/O 要求较高的应用场景。相对于 D 系列和 F 系列，Ds 和 Fs 系列的虚拟机在单位时间的使用时长上没有价格提升，但其依赖的存储服务会有所不同（详见存储计费部分）。

### 3. 操作系统的选择

由于 Windows 和大部分 Linux 发行版在使用许可上的差异，在 Azure 上，使用 Windows 操作系统的虚拟机要比使用 Linux 发行版的虚拟机，在使用成本上略高一些（目前 Azure 镜像库提供的都是免费版本的 Linux 发行版，如果用户需要在 Azure 上使用其他商业版本，需要用户拥有其合法许可）。这与在传统的采购模式中，当硬件配置相同预装 OEM Windows 的服务器，比预装 Windows 的型号的采购成本要高一些是一致的。

由于 Azure 虚拟机的应用场景主要是服务器环境，Azure 镜像库仅提供服务器版本操作系统的镜像。对于有桌面操作系统（如 Windows 7）需求的应用场景，用户可以选择上传镜像至 Azure 存储，并以此镜像创建虚拟机（用户需要先取得该版本操作系统的合法授权）。如何制作、上传镜像请参看本书的其他相关章节。

### 4. 应用负载

不仅是操作系统，用户部署在 Azure 虚拟机中的应用软件也应该是合法取得的。Azure 镜像库也提供了一些预装应用程序的操作系统镜像，例如 SQL Server，R Server，这些软件都是经过微软和其他软件开发商授权的。

用户如果使用这些镜像配置自己的虚拟机，需按虚拟机配置（CPU）和使用时长支付软件的授权费用（使用时长按虚拟机的计费时长计算）。

### 5. 负载平衡和自动缩放

负载平衡，又叫做负载均衡，是用来提高应用的用户请求响应能力，提高可用性的一种常见技术。

自动缩放，也称自动扩展，是 Azure 提供了一种能力，能根据应用的负载情况，自动增加和释放云端资源，通过配置虚拟机规模集来实现（虚拟机规模集仅适用于通过 Azure 资源管理器模型部署的虚拟机，经典部署模型的 Azure 虚拟机可使用预配置的水平或垂直自动缩放）。更多负载平衡和自动缩放的详细信息，请参看本书的相关章节。

负载平衡和自动缩放在 2 个方面影响着 Azure 虚拟机的成本：

一方面，Azure 提供了 2 个级别的虚拟机服务：基本级别和标准级别。基本级别主要用于开发、测试，或其他不要求负载平衡和自动缩放的应用场景，其价格相对低廉。标准级别的虚拟机则完整包含负载平衡和自动缩放特性，适用于更多的生产场景。

另一方面，通过使用负载平衡来提高应用的用户请求响应能力，提高应用的可用性，则需要部署 2 个或 2 个以上的 Azure 虚拟机实例。同样，配置了虚拟机规模集或水平自动缩放的应用，需要通过增加 Azure 虚拟机实例来满足应用负载的增长。增加的虚拟机实例也要按其实际使用的时长计费。虚拟机规模集在负载降低的情况下，通过减少虚拟机实例来降低成本。

6. 存储和带宽成本

Azure 虚拟机服务的持续提供，依赖于 Azure 磁盘存储服务和数据传输服务。所以这 2 项服务的全部使用量中包含用于虚拟机的部分。

7. 虚拟机的成本预估

综合考虑以上因素，可以快速评估出您所使用的 Azure 虚拟机的成本。我们以一个示例来介绍一下评估的过程：表 4.2-1 是一个传统的服务器需求，我们结合 Azure 的价格估算器来看一下，如何估算这个成本。

表 4.2-1

采购项目	配 置	操作系统	软 件	数 量
数据库服务器	CPU：双核四线程；内存：16GBSSD：128GB；硬盘：1TB	Windows Server 2012 R2	SQL Server 2012 SP2 标准版	1

第一步，我们找到 Windows 虚拟机的价格估算器，这里考虑了操作系统因素。根据需求描述中的 CPU、内存、SSD 要求，确定采用 D3 虚拟机，理由如下：

(1) 根据其 SSD 的需求，选择 D 系列；如果需要 SSD 做持久数据存储，则实际配置中需要使用 Ds 系列，其价格与 D 系列一致，评估阶段暂不做区分。

(2) CPU 方面，物理 CPU 中的一个 CPU 线程虚拟为 Azure 虚拟机中的一个 CPU 核心，双核四线程虚拟为 4 核心；如果对 CPU 主频有更高要求，则可以进一步考虑 Dv2 系列。

(3) 内存方面，由于生产工艺限制，物理内存条的容量必须以 8、16、32、64 这样的二进制整数提供，且有部分内存被 BIOS 占用，操作系统可用内存实际较低。Azure 基于虚拟化技术可以为虚拟机灵活地分配内存容量，操作系统可用内存比高，因此 14GB 虚拟内存即能提供与 16GB 内存条同等大小的可用内存空间，如图 4.2-1 所示。

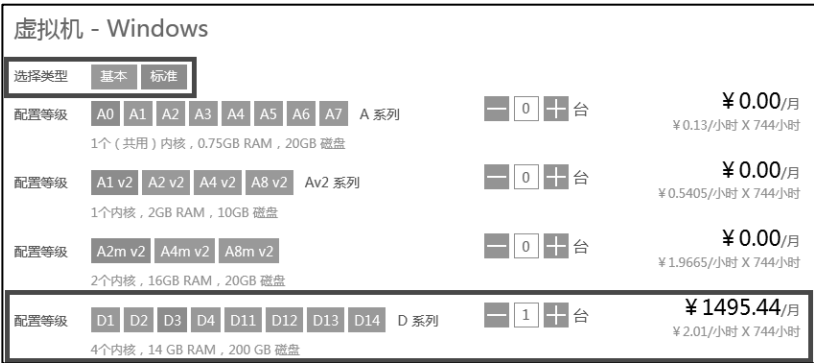


图 4.2-1

第二步，对于 SQL Server 软件的授权成本，可借助估算器选择 SQL Server 版本和虚拟机配置得出，如图 4.2-2 所示。

虚拟机 - SQL Server

选择类型

Web 版

标准版

企业版

配置等级

A0

A1

A2

A3

A4

A5

A6

A7

A 系列

0

+

台

¥0.00/月  
¥ 2.76/小时 X 744小时

适用的虚拟机实例基本或标准，1个虚拟内核

配置等级

D1

D2

D3

D4

D11

D12

D13

D14

D 系列

1

+

台

¥2053.44/月  
¥ 2.76/小时 X 744小时

适用的虚拟机实例基本或标准，4个虚拟内核

图 4.2-2

第三步，虚拟机的磁盘存储依赖于 Azure 磁盘存储服务。由于磁盘存储服务按实际使用量计费，直接在价格估算器中选择预估的使用量即可，如图 4.2-3 所示。这里的使用量应包括除虚拟机缓存盘以外的全部磁盘的使用量(含操作系统盘和用户附加的其他磁盘)。

存储 - 常规用途存储账户 - 页 Blob 和磁盘

选择类型

LRS

GRS

RA-GRS

页 blob 和  
磁盘

0

1706667

3413333

5120000

1024

GB

1

+

个

¥450.56/月

图 4.2-3

第四步，虚拟机的带宽成本也是按实际使用量计费，出站（传出数据中心）和入站（传入数据中心）分别进行累计。在本书创作时，由世纪互联运营的 Azure 每月前 1TB 的传入数据中心的数据免费，如图 4.2-4 所示。在本例中，需要配置的是数据库服务器，数据库服务器通常位于应用服务器后端，几乎没有数据通过 Internet 与数据库服务器传输，由此虚拟机产生的数据传输成本可以忽略不计。如果需要估算应用服务器与 Internet 的数据传输量，可以参考页面大小、PV（页面浏览量）、UV（单用户浏览量）等指标。由于这些参数的介绍不在本书的讨论范围，故不做介绍。

数据传输 - 传入数据中心

选择配置

0

667

1333

2000

1000

GB

0

+

个

¥0.00/月

数据传输 - 传出数据中心

选择配置

0

667

1333

2000

1000

GB

1

+

个

¥670.00/月

图 4.2-4

这台用于数据库服务器的虚拟机每月的成本消耗就是以上几个方面成本的累加。

以上仅是从单台虚拟机的维度进行成本评估，用户还可以针对其所需的各项 Azure 服务，从服务资源的维度进行综合评估。

## 4.3 存储服务的计费

### 1. 存储计费原则

Azure 提供了多种存储服务，来满足不同类型的数据存储需求。用户只需要为其使用的存储服务付费。每项存储服务的计费标准除遵循 Azure 服务的特征原则外，还具有如下特征。

#### (1) 存储与访问分开独立计算。

每个用户对 Azure 存储的需求都不一样，有的主要用于归档，较少访问这些归档的数据，但存储量较大，主要是对存储空间占用；有的虽然数据量较小，却需要频繁地访问，更多的是对单位时间内可用 I/O 的占用。

为了让用户能够更为精确地控制这些成本，Azure 采用存储与访问分开，独立计算的方式，用户仅需为其占有的存储空间和 I/O 资源付费。

与把存储、访问分开，独立计算用户的用量不同，相对传统的做法是：仅按存储容量计费，针对单位时间内用户可占有的 I/O 资源规定了上限，超过这个上限，用户就需要以更高的单位容量成本使用存储。这样的计算模型，显然不适合需要频繁访问的数据存储需求；而对于以归档为主的存储需求，用户也要承担不必要的 I/O 成本。

#### (2) 按冗余级别付费。

为保障用户位于 Azure 存储上的数据安全可靠，Azure 为用户数据提供了多重冗余。

首先，Azure 在其单个数据中心内部的多个设备之间复制用户的数据，保障了用户数据免受硬件故障（如存储设备掉电、磁盘损坏等）导致的无法读取，甚至永久丢失。

通过在同一地域内相隔数百公里的两个区域之间（在中国，这两个区域分别是距离 1000 公里以上的位于**中国东部的上海**和位于**中国北部的北京**）进行复制，避免了因为某个 Azure 数据中心的灾难性故障（例如严重性地震）带来的用户数据损失。用户可以选择某一区域作为保存其数据的主要区域，那么另一区域则是这些数据的辅助区域。

用户可以依据自身数据的可靠性要求，选择适合的冗余方案，降低其存储成本。

#### (3) 存储成本按“天”计费。

不同的数据具有不同的生命周期：财务报表可能需要长期保存；而访问日志可能仅需保留 30 天。Azure 存储以“天”而不是“月”来计算用户对存储空间的使用量，进一步降低了那些生命周期较短的数据的存储成本。例如，用户有些市场活动的海报，以 PNG 格式保存在 Azure 文件存储上，该活动将于一星期后结束。活动结束后，海报文件将从 Azure 存储上删除。假设这套海报约占 10GB 存储空间，则海报存储一周的成本约为： $1.75/31 \times 10 \times 7 = 3.95$  元，而不是  $1.75 \times 10 = 17.5$  元（Azure 网站上所示的存储价格是按月汇总后的，即假设数据存储的周期是 31 天）。

#### （4）按操作类型统计访问次数。

如上文所述，将存储与访问成本分开计算，便于用户从存储容量和 I/O 两个方面较为精确地控制其成本投入，无论是对于归档数据，还是活动数据，用户都仅需为其占用的资源付费。

而对于访问 I/O，用户可以更加精确地控制其成本，这得益于 Azure 对用户的不同 I/O 操作是分开计数并定价的。例如放置 Blob 和容器（类似文件系统创建文件和文件夹）操作，列出容器操作（类似列目录），删除操作等。这样用户就可以不断优化自身应用对存储的访问操作，来实现对存储成本的精细控制。

例如，对于后续不再使用，且无须在 Azure 存储中持久保存的数据，可以及时将其删除，这样可以在控制存储空间成本的同时，不增加 I/O 成本，因为删除操作通常是免费的。

也可以根据应用的需求，将相互关联的一组数据写入同一个 Blob，这样可以将 I/O 成本控制得更低。因为在 Blob 中写入数据，比放置一个 Blob 的成本更低。

下面我们分别介绍每种 Azure 存储服务的特点和计费逻辑。

## 2. 文件存储

Azure 文件存储使用标准的 SMB（Server Message Block）协议为应用程序和用户提供服务。SMB 是 NAS（Network Attached Storage）架构的常用协议。使用 SMB 协议的常见存储解决方案还有：Windows 文件服务器和基于 Linux 的 Samba 服务器。

像从桌面端访问典型 SMB 文件共享一样，用户可以借助 Azure 文件存储服务在应用程序的各组件之间、多个用户之间共享文件数据。

由于 Azure 将利用支持加密的 SMB 3.0 来在用户本地客户端和 Azure 文件共享之间安全地传输数据，用户本地需要使用 Windows 8/Windows Server 2012 或更新的 Windows 操作系统，以支持 SMB 3.0。如果 SMB 客户端是与 Azure 文件共享在同一区域中的 Azure 虚拟机，则其他支持 SMB 2.1 的操作系统也受支持。关于各 Windows 版本支持的 SMB 版本，可参考 <https://blogs.technet.microsoft.com/josebda/2013/10/02/windows-server-2012-r2-which-version-of-the-smb-protocol-smb-1-0-smb-2-0-smb-2-1-smb-3-0-or-smb-3-02-are-you-using/>。

与其他 Azure 服务一样，用户也可以通过文件存储 API 来访问共享中的文件数据。

支持通用的 SMB 存储协议，使 Azure 文件存储服务具有更好的兼容性，方便用户应用的逐步云化需求。

由世纪互联运营的 Azure 文件存储目前提供 2 个冗余级别的服务：本地冗余（LRS）和异地冗余（GRS）。

其存储空间成本分别按各级别的实际使用量计算，最小计费单位为“/GB/天”。

访问成本分为 4 类操作分别进行统计，分别是：

- 放置文件、创建容器操作（按 10 000 计）
- 列出容器操作（按 10 000 计）
- 除删除（此操作免费）之外的其他操作（按 10 000 计）
- 文件协议操作（按 10 000 计）



### 3. 表存储

Azure 表存储服务为半结构化数据的存储提供了基础架构。所谓半结构化数据，不同于传统数据库的二维表，数据库二维表的每行拥有相同的字段，而在半结构化数据中，代表现实实体的行可以拥有不同数量，不同名称的字段，称为属性。如下面这个案例：

公司人力资源部门保存的员工档案中，每个员工都有相同的属性，如姓名、出生日期、入职日期、岗位、基本薪金等；也有些属性只是部分员工才拥有的，如项目经验对于行政员工则不具备。如果仅是如此，仍然比较适合在传统的数据库中保存，对于不需要项目经验的员工，该字段留空即可。

如果行政员工恰巧也有一些字段是项目工程师不具备的，则我们要在表结构中再增加一些字段，对于项目工程师，该字段留空。如果这样的情况再复杂一些，多个岗位都有其独有的字段，那么这个表的结构将变得相当松散。如果某些项目工程师的项目经验比较多，则在表结构的设计上和查询逻辑的编写上就相对麻烦些。

而 Azure 表存储的每个实体可以拥有不同数量、不同字段名称的属性，上面案例中的情况则非常适合用 Azure 表存储来进行数据管理。

在传统环境中，针对这种半结构化的数据，比较常用的解决方案是 XML 和 JSON，而 Azure 表存储服务与 JSON 有很好的互操作性，无疑为用户的半结构化数据基础结构向云端扩展提供了更好的选择。

由世纪互联运营的 Azure 表存储目前提供 3 个冗余级别的服务，分别是：本地冗余（LRS）、异地冗余（GRS）和读取访问跨异地冗余（RA-GRS）。

与标准的异地冗余级别不同，这种冗余级别支持在对主要区域进行读写的同时，对辅助区域的数据提供只读访问，更适合“一地写入，多地访问”的应用场景。标准异地冗余中的辅助区域仅供灾备使用。

表存储的存储成本分别按各级别的实际使用量计费，并按用量“阶梯定价”，最小计费单位为“/GB/天”。具体价格见 Azure 网站的价格页面。

表存储的访问成本分为 6 类操作分别进行统计，分别是：

- 放置表、创建容器操作（按 10 000 计）
- 批处理放置表、创建容器操作（按 10 000 计）
- 列出容器操作（按 10 000 计）
- 表读取操作（按 10 000 计）
- 扫描容器操作（按 10,000 计）
- 表删除操作（按 10,000 计）

### 4. 队列存储

Azure 队列存储是一种云端的消息服务。消息服务是现代分布式应用结构中常见的组成部分，用来在各个应用组件之间传递所需信息。很多软件厂商都有用于本地化部署的消息服务软件。

图 4.3-1 展示了消息服务在订单系统中的应用场景，在未引入消息中间件时，用户的每次下单都要发起对库存组件的调用，下单组件收到库存组件的返回值，则订单成功。如

果因为网络等原因未收到该返回值，则订单失败。引入了消息中间件之后，订单组件将订单状态写入消息中间件，完成订单；库存组件从消息中间件中读取订单状态，其他组件也可以从消息中间件读取订单状态，实现了订单状态在各组件之间共享，各组件也可以独立运行，处理过程无须等待其他组件，减少了每个订单的处理时间。

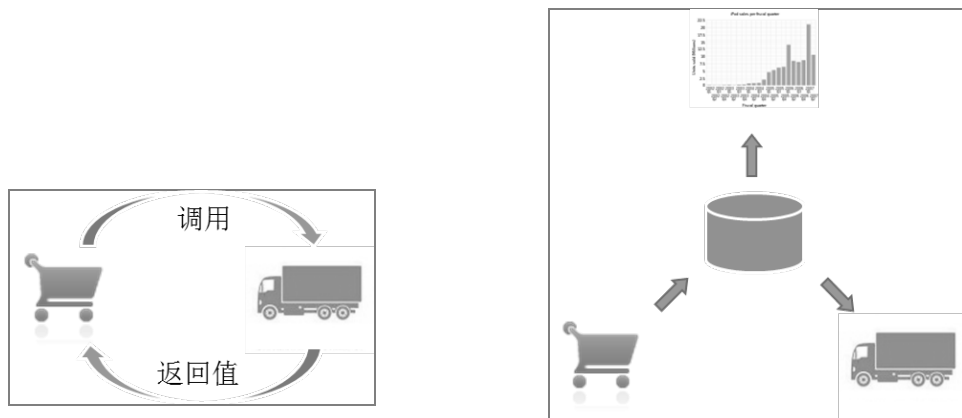


图 4.3-1

Azure 同时提供另一项消息队列的服务，叫做服务总线。两者的对比可以参考：<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-azure-and-service-bus-queues-compared-contrasted>。

由世纪互联运营的 Azure 队列存储目前提供 3 个冗余级别的服务，分别是：本地冗余（LRS）、异地冗余（GRS）和读取访问跨异地冗余（RA-GRS）。

其存储成本分别按各级别的实际使用量计费，并按用量“阶梯定价”，最小计费单位为“/GB/天”。具体价格见 Azure 网站的价格页面。

队列存储的访问成本即是队列的操作：队列操作（按 10 000 计）。

## 5. 磁盘存储/页 Blob

Azure 页 Blob 与块 Blob 和 Azure 文件存储一样，可以通过互联网存取较大的二进制非结构化数据。

页 Blob 更为随机存取做了优化，适合用来存储虚拟磁盘文件，以应对传统操作系统对文件系统的随机读写。事实上，Azure 虚拟机的虚拟磁盘文件（VHD）正是存储于 Azure 页 Blob 服务。正如前文提到的，Azure 虚拟机依赖于某项 Azure 存储服务，除用于缓存的磁盘外，Azure 虚拟机的磁盘都是由页 Blob 提供的存储服务，所以页 Blob 又称为磁盘存储。

由世纪互联运营的 Azure 磁盘存储目前提供 2 个性能级别的服务，分别是标准磁盘和高级磁盘。高级磁盘服务由 SSD 提供，具有非常高的吞吐量和低延迟，旨在支持 I/O 密集型工作负荷。高级磁盘进一步提供了 3 个子级别的服务，提供不同的容量和性能，分别是 P10，P20，P30。

在冗余级别上，标准磁盘服务提供 3 个冗余级别：本地冗余（LRS）、异地冗余（GRS）和读取访问跨异地冗余（RA-GRS）；高级磁盘服务目前仅提供本地冗余级别。

标准磁盘服务的存储成本分别按各冗余级别的实际使用量计费,并按用量“阶梯定价”,最小计费单位为“/GB/天”。具体价格见 Azure 网站的价格页面。

标准磁盘服务的访问成本分为 3 类操作分别进行统计,分别是:

- 放置页 Blob、创建容器操作(按 10 000 计)
- 列出容器操作(按 10 000 计)
- 除删除(此操作免费)之外的其他操作(按 10 000 计)

高级磁盘的计量单位比较特殊,为“/磁盘/小时”。例如,如果在设置完 P10 磁盘的 20 小时后删除它,则会以 20 小时计算 P10 磁盘的费用。这与写入磁盘的实际数据量或使用的 IOPS/吞吐量无关。

## 6. 块 Blob

块 Blob 为通过互联网存取较大的二进制非结构化数据而设计。使用 Azure Blob API,用户应用可以将需要上传的数据分成大小不同的块,多个块并行传输,提高传输效率。对于传输失败的块,也只需要重传这个块,无须重传全部数据。多线程传输和断点续传这样的功能可以很容易地在用户的应用中实现。

由世纪互联运营的 Azure 块 Blob 存储目前提供 3 个冗余级别的服务,分别是:本地冗余(LRS)、异地冗余(GRS)和读取访问跨异地冗余(RA-GRS)。

每个冗余级别还提供“冷”和“热”2 个访问层,其分别对应的是**低廉的存储成本,但访问成本略高;低廉的访问成本,但存储成本略高**。用户可以选择“冷”存储保存归档数据,选择“热”存储保存需要频繁访问的数据,以分别降低每种数据的总体成本消耗。

块 Blob 的存储成本分别按各级别的实际使用量计费,并按用量“阶梯定价”,最小计费单位为“/GB/天”。具体价格见 Azure 网站的价格页面。

块 Blob 存储的访问成本分为 5 类操作分别进行统计,分别是:

- 放置 Blob 或块、列出并创建容器操作(按 10 000 计)
- 除删除(此操作免费)之外的其他操作(按 10 000 计)
- 数据检索(GB)
- 数据写入(GB)
- 异地复制数据传输(GB)

## 4.4 网络资源的计费

对 Azure 用户来说,Azure 提供的网络资源包括公共 IP 地址、各种虚拟网络设备,以及网络负载等。下面我们分别介绍一下 Azure 对这些网络资源的计费。

### 1. 公共 IP 地址

众所周知,公共 IPv4 地址空间是有限的,为提高 IPv4 地址的使用效率,Azure 默认的 IPv4 地址分配原则是动态。即在服务(如 Azure 虚拟机)启动的时候为其分配公共 IPv4 地址,在停止时释放。这样,在不同时间,多个应用(可能分属不同的用户)可以共享这个 IPv4 地址。服务在重新启动时,可能被分配到与之前不同的 IPv4 地址。使用动态分配的公共 IPv4 地址也是受到鼓励的,Azure 不会向用户收取费用。

Microsoft Azure 暂不支持 IPv6，未来将提供支持。另外私有 IP 地址空间不要求全局唯一性，不属于有限资源，所以以下提到的 IP 地址如无特殊说明，均指公共 IPv4 地址。

Azure 也允许用户付费使用相对固定的公共 IP 地址，以适应某些应用的需要，或其他相关要求。这时 IP 地址是在服务被创建时分配，在删除时释放，用户的服务在其存在的生命周期内，即使中途停止也不会改变其分配到的 IP 地址。

公共 IP 地址按使用时长计费，与 Azure 虚拟机以分钟为单位计量不同，公共 IP 地址以小时为基本单位，对于不足 1 小时的使用量按 1 小时计算。

因为 Azure 目前有 2 种资源/服务管理模型：Azure 资源管理器模型（Azure Resource Manager）和 Azure 服务管理模型（Azure Service Management，又称经典部署模型）。关于这两种模型的对比，请参考本书相关章节。

公共 IP 地址在 Azure 服务管理模型中的使用和计费情况略复杂些，具体分为以下情况。

（1）分配给“云服务”（负载均衡器）的虚拟 IP 地址（VIP）。

Azure 服务管理模型中有一个“云服务”的概念，是按用户需求组织起来的一“台”或多“台”虚拟机，以及一“台”虚拟负载均衡器，如图 4.4-1 所示。

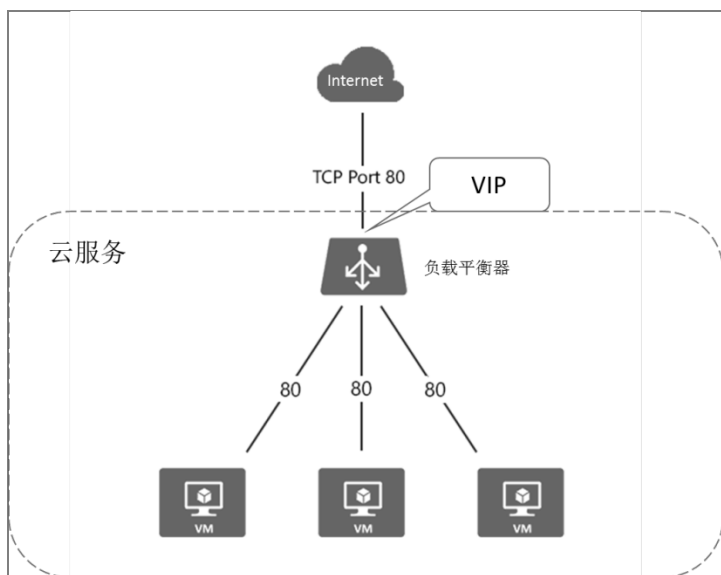


图 4.4-1

公共 IP 地址默认被分配给整个“云服务”，实际是配置在负载均衡器的入站接口，称为虚拟 IP 地址（VIP）。该 IP 地址默认是动态分配，即当“云服务”中的第一“台”虚拟机启动时分配，当“云服务”中的虚拟机全部停止时释放。这时公共 IP 地址是免费提供的，仅当下面的情况发生时开始计费。

（2）附加的 VIP 地址。

Azure 默认只为每个“云服务”/负载均衡器分配一个 VIP 地址，如果用户有多个 IP 地址的需求，则需要附加 VIP 地址。如下面的场景：

用户有 2 个或 2 个以上的独立网站，需要各自使用不同的公共 IP 地址，每个网站的负

载不是很高，但却具有比较高的可靠性，需要至少 2 台虚拟机运行，避免虚拟机故障导致网站服务中断。这与传统的“虚拟主机”业务类似。

这时，对于用户附加的 VIP 地址，则需按使用时长计费，以小时为单位，不足 1 小时按 1 小时计算。

### （3）实例级公共 IP 地址（PIP）。

VIP 实际配置在负载均衡器的进站接口，而不是虚拟机的网络接口。用户可以为某些虚拟机直接分配公共 IP 地址，称为实例级公共 IP 地址（PIP）。不可直接从虚拟机 guest 操作系统中配置 PIP，具体配置方法可参考 Azure 网站文档。

实例级公共 IP 地址需要按使用时长计费，以小时为单位，不足 1 小时按 1 小时计算。

### （4）保留的虚拟 IP 地址。

“云服务”的虚拟 IP 地址默认是动态分配的，也可以配置为相对固定的使用方式。这在 Azure 服务管理模型中是通过保留实现的，称为保留 IP 地址。

用户需要先从 Azure 可用的地址池中申请保留地址，Azure 会为客户保留指定区域的公共 IP 地址，避免分配给其他用户，随后用户可以按自己的需要将该地址关联给指定的“云服务”，或重新映射给其他的“云服务”。

保留的 IP 地址可以免费使用，但以下行为是收费的：

- a. 只保留，不使用。即保留的 IP 地址没有被关联到任何“云服务”，那么需要按保留时长计费，以小时为单位，不足 1 小时按 1 小时计算。
- b. 保留过多的公共 IP 地址。不建议用户过多地使用保留地址，而是使用默认的 VIP，通过域名来访问自己的应用，以节省有限的公共 IPv4 地址资源。Azure 允许用户免费使用的保留地址最多为 5 个，如果用户保留超过 5 个公共 IP 地址，从第 6 个起，即使已经关联到特定的“云服务”，仍然需要按保留时长计费，以小时为单位，不足 1 小时按 1 小时计算。
- c. 频繁变更使用对象。用户可以将已保留的 IP 地址从当前的“云服务”中“删除”，重新关联到其他“云服务”。Azure 允许用户每个月可以改变这种映射关系 100 次，超过则需要按次数计费。

我们可以看出，Azure 鼓励所有用户分时复用公共 IPv4 地址；鼓励使用域名（而不是 IP 地址）唯一地标识其主机。用户可以在这些受鼓励的场景下，免费地使用公共 IP 地址。

Azure 也允许用户通过静态分配、保留等手段，独立地使用公共 IP 地址，以满足各种应用场景的需求。

对于比较高的公共 IPv4 地址消耗，需要用户承担一定的成本，来满足某些特殊的应用场景。

## 2. 各种虚拟网络设备

Azure 引入了多种虚拟网络设备的概念，来满足以下需求：

- a. Azure 虚拟机之间可以通信；
- b. Azure 虚拟机与外界环境可以通信；
- c. 按用户的需要（例如隔离，加密，负载平衡等）进行通信；
- d. 以用户熟悉的方式（概念，工具等）管理通信。

这些虚拟的网络设备实际消耗的是计算资源,基本的计量标准是计算资源的使用时长。因为这些虚拟设备是为用户的网络而服务的,故我们将其视为一种网络资源。

下面我们分别介绍一下常见的虚拟网络设备的计费情况。

#### (1) 虚拟的网络交换机。

Azure 引入了“虚拟网络”的概念,可以为“虚拟网络”中的虚拟机提供二层连接,广播域隔离, DHCP 等服务。为了便于理解,我们可以想象“虚拟网络”中有一“台”“虚拟交换机”。

这些服务在 Azure 上免费提供,并且虚拟机通过虚拟网络的内部通信量也不会对用户收费。

#### (2) 虚拟的 VPN 网关设备。

用户可以在 Azure 虚拟网络中创建虚拟的 VPN 网关设备,与用户位于其他位置的 VPN 网关一起,建立 VPN 隧道(站点到站点 VPN)。这样, Azure 虚拟网络中的虚拟机可以跨越 Internet 加入用户的私有网络。对端的 VPN 网关可以是物理设备、其他解决方案中的虚拟设备,或其他 Azure 区域的虚拟 VPN 网关。

用户也可以配置 Azure 虚拟网络中的 VPN 网关,让分散在 Internet 上的授权用户通过 VPN 客户端拨入到 Azure 虚拟网络中来(点到站点 VPN),让出差和在家办公的用户可以接入 Azure 虚拟网络。

由世纪互联运营的 Azure 目前提供 3 个级别的 VPN 网关:基本 VPN 网关、标准 VPN 网关、高性能 VPN 网关,分别提供不同的吞吐量和最大支持的隧道数。VPN 网关按使用时长计费,以小时为单位,不足 1 小时按 1 小时计算。

#### (3) 专线路由器。

如果用户对 Azure 虚拟网络的私有连接要求更高的网络性能保障,可以向与 Azure 合作的电信运营商申请“专线”服务,在 Azure 虚拟网络中建立虚拟路由器,称为 ExpressRoute,与“专线”另一端的支持 BGP 的路由器建立三层通信。ExpressRoute 连接并不通过 Internet,与通过 Internet 的 VPN 相比,可靠性更高、速度更快、延迟时间更短。

由世纪互联运营的 Azure 目前提供 4 个级别的 ExpressRoute 网关:基本 ExpressRoute 网关、标准 ExpressRoute 网关、高性能 ExpressRoute 网关和超高性能 ExpressRoute 网关,其支持的最大吞吐量不同。ExpressRoute 网关的价格与同级别的 VPN 网关相同,超高性能 ExpressRoute 价格更高一些。

“专线”的相关费用请见下文的网络负载部分。

#### (4) 负载均衡。

Azure 提供多个级别的负载均衡,分别是:

- Azure 负载均衡器

Azure 负载均衡器免费提供,为虚拟机提供四层的负载平衡。在 Azure 服务管理部署模型中的“云服务”中,默认即包含此负载均衡器。

- 应用程序网关

Azure 应用程序网关是七层(HTTP/HTTPS)负载均衡设备,目前提供 3 个不同等级的服务,由不同配置的虚拟机提供。应用程序网关费,以小时为单位,不足 1 小时按 1 小时计算。

每个等级的应用程序网关包含数量不同的免费数据处理，例如，大型应用程序网关每月可以免费处理 40TB 的数据，超过 40TB 的数据需要按 GB 计费；中型应用程序网关每月可以免费处理 10TB 的数据，超过 10TB 的数据需要按 GB 计费；小型应用程序网关则没有免费的数据处理，全部处理数据都需要按 GB 计费。具体计费标准请参见 Azure 价格介绍页面。

#### ● 流量管理器

在传统负载均衡的应用场景中，多个资源主机通常位于同一数据中心，资源主机与负载均衡器之间具有高速内部链路，Azure 负载均衡器与 Azure 应用程序网关提供的都是此类的解决方案。

与传统负载均衡面向的应用场景不同，Azure 流量管理器所管理的 HTTP 资源分布在互联网上，可以是 Azure 虚拟机、Azure Web 应用、其他 Web 站点等。资源之间没有高速链路，但某个资源提供点可能与特定群体的最终用户之间拥有最优链路，如部署在教育网内的服务节点比较适合供教育网用户优先访问。Azure 流量管理器基于 DNS 查询在多个 HTTP 资源之间进行负载均衡，更多技术细节请参考本书相关章节。

流量管理器作为 DNS 系统的扩展，根据 HTTP 源的健康状况动态地向 DNS 系统返回不同的查询结果，是一项 PaaS 服务。Azure 用户并不独占流量管理器的资源，仅当 DNS 系统（最终用户）向流量管理器查询属于 Azure 用户的域名时，及流量管理器为 Azure 用户的 HTTP 资源做健康检查时，才会消耗流量管理器的资源。所以对用户的收费要计算流量管理器查询用户域名的次数和用户需要检查的 HTTP 源的数量。具体计费标准可参考 Azure 流量管理器的价格详情页面，或向 Azure 客服中心咨询，这里不再赘述。

### 3. 网络负载

用户的网络负载在 Azure 网络基础架构的多个大小不同的范围传输，是对 Azure 数据中心的各级网络带宽资源的利用。Azure 针对不同用户的不同负载情况，公平地按使用量计费，单位是 Giga Bytes (GB)。以下是各级网络负载的计费说明：

#### (1) 虚拟网络内部的网络负载。

在前面介绍虚拟网络的时候提到，Azure 不会针对用户部署在同一虚拟网络中的虚拟机之间的数据传输收费。

#### (2) 虚拟网络之间的网络负载。

位于同一 Azure 区域的 2 个虚拟网络可以配置对等互联，实现三层通信。对等互联的 2 个虚拟网络需要按入站和出站分别计算使用流量，并收取很少的费用。

**Tips:** 建立对等互联的 2 个虚拟网络可以来自同一个 Azure 订阅，也可以来自不同的 Azure 订阅。

**Tips:** 建立对等互联的 2 个虚拟网络至少有 1 个是基于 Azure 资源管理器模型构建的。

#### (3) 不同 Azure 区域之间的网络负载。

用户可以利用 Azure 区域间的专线连接，在位于 2 个 Azure 区域的虚拟网络之间建立 VPN，实现跨区域的内部连接。连接 VPN 的 2 个虚拟网络需要分别计算各自的出站流量，并收取少量的费用。

（4） Azure 数据中心与 Internet 的网络负载。

更多典型的网络负载发生在 Azure 数据中心与 Internet 之间，其利用的是 Azure 数据中心的互联网接入带宽。目前，由世纪互联运营的 Azure 需要分别计算用户的上行和下行流量，并收取相应的费用。用户每月产生的前 1TB 入站流量免费（世纪互联随时有可能调整此价格和优惠）。

很多 Azure 服务的使用都要通过互联网传输数据到 Azure 数据中心，都要计算入站和出站流量。如果把互联网接入也看做是一项 Azure 服务的话，那么该服务是需要被包括虚拟机在内的多项 Azure 服务依赖的服务。如果您的预计流量较大，那么您需要把数据传输成本考虑在内。数据传输流量的预估可以参考页面大小、PV（页面浏览量）、UV（单用户浏览量）、下载量等指标，不再赘述。

（5） Azure 数据中心通过专线与客户网络传递的网络负载。

用户使用与 Azure 合作的电信运营商的“专线”，将其广域网络拓展到 Azure。其“专线”服务有 2 种计费模式：

- 无限数据计划

采用无限数据计划，用户仅需按端口速度支付固定的月度费用，无流量限制，如端口速度为 50Mbps 的“专线”每月价格为 2815.12 元，适合流量较大的专线用户。

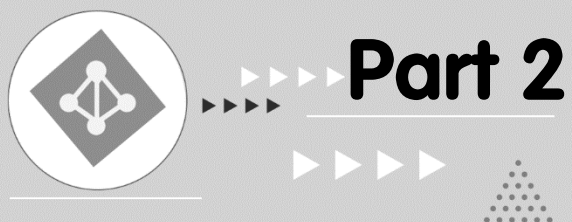
- 计量数据计划

采用计量数据计划，用户仅需按端口速度为其“专线”支付较低的固定费用，然后按出站流量计算使用量。这种模式适合传输少量数据的专线用户。

注：

- （1） 用户如需与 Azure 建立“专线”连接，ExpressRoute 网关的费用也是必须的。
- （2） 电信运营商可能向用户收取其他费用，具体请与电信运营商咨询。





# 技术部分

---

## 第五章 计算与云服务

本章详细介绍了各种不同型号虚拟机的特点和应用，磁盘和映像的概念和使用方法，Linux 不同发行版的图形化的配置方法，虚拟机扩展的原理与使用，以及多网卡虚拟机的配置步骤。同时，也对云服务相关的配置和注意事项做了介绍，例如云服务 DNS 解析，实例级公共 IP 地址的 DNS 绑定，反向 DNS 解析以及多 VIP 的使用参考等，便于读者更加深入地了解虚拟机和云服务的配置和部署。

### 5.1 虚拟机的使用简介

#### 5.1.1 A 系列虚拟机

A 系列虚拟机是最基本也是使用最广泛的一类虚拟机，目前 Azure 上支持的 A 系列虚拟机分为基本（Basic）和标准（Standard）两种。

基本级的虚拟机大小与同等的标准级大小具有类似的配置，价格要低于同等的标准级虚拟机，但它不具备 Azure 负载平衡器和自动伸缩功能，所以通常使用标准级虚拟机进行生产环境的部署和配置，基本级虚拟机建议仅作为测试使用。

A 系列标准级虚拟机的配置参数可以参考表 5.1-1。

表 5.1-1

型 号	CPU 核数	内存/GB	本地磁盘 /GB	最大数据 磁盘数	数据磁盘最大 吞吐量/IOPS	最大网卡数/网络带宽 等级
Standard_A0	1	0.768	20	1	1×500	1/低
Standard_A1	1	1.75	70	2	2×500	1/中
Standard_A2	2	3.5	135	4	4×500	1/中
Standard_A3	4	7	285	8	8×500	2/高
Standard_A4	8	14	605	16	16×500	4/高
Standard_A5	2	14	135	4	4×500	1/中
Standard_A6	4	28	285	8	8×500	2/高
Standard_A7	8	56	605	16	16×500	4/高

要使用 Azure 虚拟机，可以通过经典管理门户或者新门户进行创建，也可以通过 Azure Powershell/Cli 或者其他 SDK 进行创建。以下为使用经典管理门户创建虚拟机的操作步骤：

登录到经典管理门户，在窗口底部命令栏中单击“新建”，依次选择“计算”——“虚拟机”——“从库中”，如图 5.1-1 所示。



图 5.1-1

在弹出的对话框中，第一步要进行映像或磁盘的选择，如图 5.1-2 所示。



图 5.1-2

在左侧的 MICROSOFT 下方是 Microsoft 提供的 Windows 相关的平台映像，在左侧选择不同类型的映像，中间可以选择不同的版本和特性的映像，右侧则是关于映像的介绍。

在左侧还分别列出了 UBUNTU, COREOS, CENTOS, SUSE 等开源 Linux 映像可供选择。

如果要选择自定义上传或者用户自己捕获的映像，可以在“我的映像”中找到。如果希望通过已有的操作系统磁盘来创建虚拟机，可以在“我的磁盘”中找到对应磁盘进行创建。

选择一个映像后，单击“下一步”，进入到虚拟机的基本配置界面，如图 5.1-3 所示。

创建虚拟机

### 虚拟机配置

版本发布日期 ?  
2016/12/14

虚拟机名称 ?  
DemoVM

层  
基本 标准

大小 ?  
A3 (4 核, 7 GB 内存)

新用户名  
demouser

新密码 确认  
.....

**Windows Server 2012 R2 Datacenter (en-us)**

At the heart of the Microsoft Cloud OS vision, Windows Server 2012 R2 brings Microsoft's experience delivering global-scale cloud services into your infrastructure. It offers enterprise-class performance, flexibility for your applications and excellent economics for your datacenter and hybrid cloud environment. This image includes Windows Server 2012 R2 Update.

OS 系列  
Windows

发布者  
Microsoft

磁盘数  
1

位置  
China East+China North

定价信息  
定价根据你选择用于设置虚拟机的订阅而异。

图 5.1-3

可以在此界面调整所选映像的发布版本，默认会选择发布日期最新的版本。

这里配置的虚拟机名称不同于主机名，这个名称用于显示在虚拟机列表中，不可更改，所以建议选择一个更为贴近虚拟机使用目的的名称。虚拟机名称只能包含字母、数字和连字符，它还必须以字母开头，并以字母或数字结尾。

虚拟机的大小会影响其使用成本，还会影响某些配置选项，例如，可以附加的数据磁盘数，具体请参阅表 5.1-1。虚拟机大小配置可以在后期进行调整，为了尽可能让虚拟机部署在较新的区域内，可以在开始创建时使用较新的类型（例如 D 系列或 F 系列的），便于后期将虚拟机提升到更高的型号。

“新用户名”指用于管理服务器的管理账户，密码的长度必须是在 8~123 个字符之间，并且必须至少包含这四种字符中的三种：小写字符、大写字符、数字和特殊字符（标点符号）。除了使用用户名和密码的方式登录到虚拟机之外，对于 Linux 虚拟机，也可以以配置密钥的方式登录。新用户名不能以 administrator, admin, test, root 等词作为值，避免被暴力破解的可能。

完成上述基本配置后，单击“下一步”进入虚拟机详细配置界面，如图 5.1-4 所示。

虚拟机可以选择创建到新的云服务下，如果选择“创建新云服务”，则 Azure 会自动创建一个与该虚拟机名称相同的云服务，并将此虚拟机创建到该云服务下。也可以在下拉列表中选择“已有云服务”，将虚拟机创建到其中。

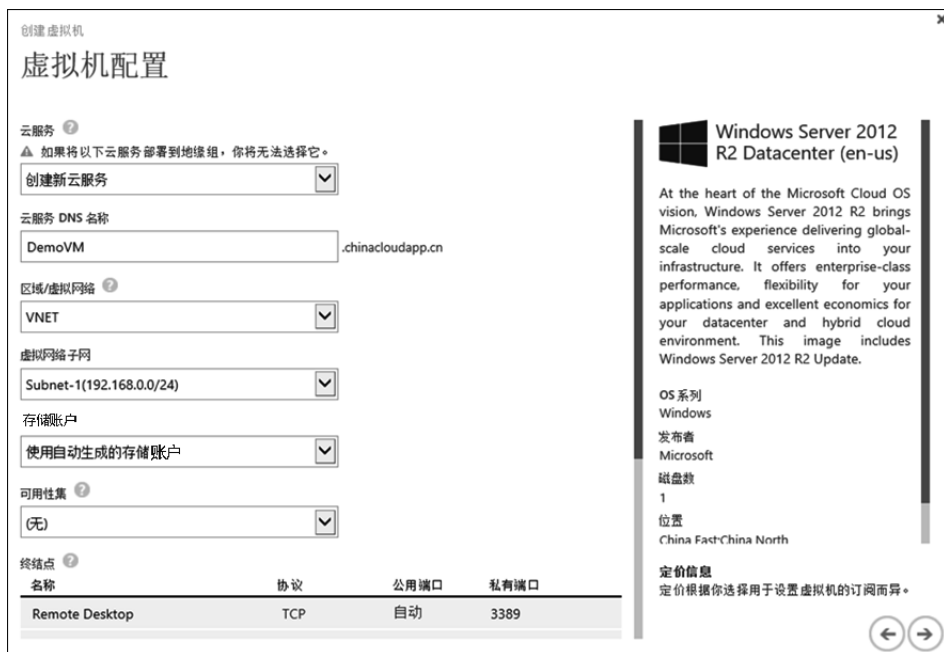


图 5.1-4

如果要创建新的云服务，则需要为云服务指定一个 DNS 名称，这个 DNS 名称在公网上必须是唯一的，Azure 会将其指向云服务的公网 IP（A 记录）。该 DNS 名称创建后不可修改。

在“区域/地域组/虚拟网络”中选择适合部署所在位置的区域（中国东部/中国北部），也可以选择指定一个虚拟网络，目前 Azure 中已经使用区域虚拟网络取代了传统的地缘组，因此强烈建议使用虚拟网络而非地域组来部署环境。虚拟机在创建完成后，不能再通过配置修改进行虚拟网络或者区域的变更。

“存储账号”用于存放虚拟机的持久化磁盘，如果选择“使用自动生成的存储账户”，则 Azure 会在虚拟机创建前，随机计算一个散列值并将其作为名称创建一个新的存储账户，由于随机生成的散列值名称无实际意义，不便于记忆，因此建议提前创建一个名称具有实际意义的存储账号，然后在这里选择创建好的存储账号名称。

“可用性集”用于虚拟机高可用性的配置，同一云服务下的虚拟机可以加入同一可用性集中，在同一可用性集中的虚拟机的更新域和故障域不同，可以最大程度地避免同一时间维护或宕机。关于故障域和更新域的具体说明，请参考链接：<https://www.azure.cn/documentation/articles/azure-iaas-user-manual-part1/>。

“终结点”用于将云服务的公网 IP 地址与公共端口号映射到虚拟机的内网地址与内网服务侦听端口号。

配置完成后，单击“下一步”，进入安装虚拟机代理（Agent）和扩展的配置界面，如图 5.1-5 所示。

虚拟机代理为你提供安装扩展的环境，可帮助你与虚拟机交互或管理虚拟机，关于虚拟机代理以及扩展，会在后续章节详细介绍。



图 5.1-5

配置全部完成后，单击“√”，虚拟机会完成创建，创建后的虚拟机将显示在经典管理门户虚拟机列表中。如果新创建了云服务 and 存储，相应的云服务和存储账户会在虚拟机创建前完成创建，并将其列在这些部分中。虚拟机和云服务都会自动启动，其状态将显示为“正在运行”。

### 5.1.2 D 系列虚拟机

对于需要更快速的本地（临时）存储和更快的 CPU 的用户来说，D 系列虚拟机提供了更卓越的性能。D 系列虚拟机最多可以提供 112 GB 内存，计算处理器速度比 A 系列虚拟机快约 60%。此外，D 系列虚拟机最多还可以提供 800 GB 的本地 SSD 磁盘空间，可以瞬间实现读取和写入操作。对于正在运行的生产系统，如果需要增加处理能力并快速进行本地磁盘 I/O，D 系列虚拟机也是不错的选择。

D 系列虚拟机的配置参数可以参考表 5.1-2 和表 5.1-3。

表 5.1-2

型 号	CPU 核数	内存/GB	本地磁盘(SSD) /GB	最大数据 磁盘数	数据磁盘最大 吞吐量/IOPS	最大网卡数/网络 带宽等级
Standard_D1	1	3.5	50	2	2×500	1/中
Standard_D2	2	7	100	4	4×500	2/高
Standard_D3	4	14	200	8	8×500	4/高
Standard_D4	8	28	400	16	16×500	8/高

表 5.1-3

型 号	CPU 核数	内存/GB	本地磁盘 (SSD) /GB	最大数据磁 盘数	数据磁盘最大 吞吐量/IOPS	最大网卡数/网 络带宽等级
Standard_D11	2	14	100	4	4×500	2/高
Standard_D12	4	28	200	8	8×500	4/高
Standard_D13	8	56	400	16	16×500	8/高
Standard_D14	16	112	800	32	32×500	8/极高

在这些型号中，本地磁盘（即临时磁盘，对于 Windows 虚拟机，默认盘符为 D 盘；对于 Linux 虚拟机，默认会挂载到/mnt 或/mnt/resource 下）是本地 SSD。

这种高速本地磁盘的最佳使用场景包括：将工作负载同步到多个实例，例如 MongoDB；或者将这种高速 I/O 的磁盘用于本地和临时缓存，配置例如 SQL Server 2014 的缓冲池扩展。

需要注意的是，临时磁盘并非持久化的数据磁盘，因此，当虚拟机由于硬件故障或者释放资源而迁移到其他物理主机时，临时磁盘的数据就会丢失。这与系统磁盘以及任何附加的数据磁盘不同，系统磁盘和附加的数据磁盘是在 Azure 存储中的持久化磁盘。

### 5.1.3 F 系列虚拟机

F 系列基于 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell)处理器，该处理器使用 Intel Turbo Boost 2.0 技术，可实现高达 3.1 GHz 的时钟速度。F 系列虚拟机的 CPU 性能与 Dv2 系列虚拟机的相同。根据每个核心的 Azure 计算单位（ACU），F 系列在 Azure 产品组合中具有最高性价比。

F 系列虚拟机的配置参数可以参考表 5.1-4。

表 5.1-4

型 号	CPU 核数	内存/GB	本地磁盘 (SSD) /GB	最大数据 磁盘数	数据磁盘最大 吞吐量/IOPS	最大网卡数/网络 带宽等级
Standard_F1	1	2	16	2	2×500	1/中
Standard_F2	2	4	32	4	4×500	2/高
Standard_F4	4	8	64	8	8×500	4/高
Standard_F8	8	16	128	16	16×500	8/高
Standard_F16	16	32	256	32	32×500	8/极高

F 系列虚拟机型号的命名引入了对 Azure 虚拟机型号命名的新标准。对于 F 系列和未来发布的其他系列的虚拟机的型号名称，主名称字母后的数字与 CPU 内核数量相匹配。

### 5.1.4 DS/FS 系列虚拟机

DS/FS 系列虚拟机的计算性能分别与 D 系列和 F 系列的虚拟机对应型号相同，区别在于 DS 和 FS 系列虚拟机可使用高级存储，从而为 I/O 密集型应用提供高性能、低延迟的存储。关于高级存储的使用方法，可以参考后续存储的相关章节。

DS 系列虚拟机不同型号对应的磁盘吞吐量相关指标可参考表 5.1-5（其他指标请参考同型号 D 系列虚拟机）。

表 5.1-5

型 号	缓存磁盘最大吞吐量：IOPS / MBps (以 GB 为单位的缓存大小)	非缓存磁盘最大吞吐量：IOPS / MBps
Standard_DS1	4000/32(43)	3200/32
Standard_DS2	8000/64(86)	6400/64
Standard_DS3	16000/128(172)	12800/128
Standard_DS4	32000/256(344)	25600/256
Standard_DS11	8000/64(72)	6400/64
Standard_DS12	16000/128(144)	12800/128
Standard_DS13	32000/256(288)	25600/256
Standard_DS14	64000/512(576)	51200/512

FS 系列虚拟机不同型号对应的磁盘吞吐量相关指标可参考表 5.1-6（其他指标请参考同型号 F 系列虚拟机）。

表 5.1-6

型 号	缓存磁盘最大吞吐量：IOPS / MBps (以 GB 为单位的缓存大小)	非缓存磁盘最大吞吐量：IOPS / MBps
Standard_F1s	4000/32(12)	3200/48
Standard_F2s	8000/64(24)	6400/96
Standard_F4s	16000/128(48)	12800/192
Standard_F8s	32000/256(96)	25600/384
Standard_F16s	8000/64(192)	6400/768

5.1.5 2 代虚拟机

2 代虚拟机是基于最新一代的 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell)处理器，该处理器使用 Intel Turbo Boost 2.0 技术，可实现高达 3.1 GHz 的时钟速度。对于对计算资源需求较高的应用来说，2 代虚拟机提供了最佳的处理器性能。

目前推出的 2 代虚拟机主要有 Av2、Dv2、DSv2（磁盘吞吐量性能可参考同型号 DS 虚拟机数据）三个系列，主要性能指标如表 5.1-7～表 5.1-9 所示。

表 5.1-7

型 号	CPU 核数	内存/GB	本地 SSD 磁盘 /GB	最大数据 磁盘数	数据磁盘最大 吞吐量/IOPS	最大网卡数/网络 带宽等级
Standard_A1_v2	1	2	10	2	2×500	1/中
Standard_A2_v2	2	4	20	4	4×500	2/高
Standard_A4_v2	4	8	40	8	8×500	4/高
Standard_A8_v2	8	16	80	16	16×500	8/高
Standard_A2m_v2	2	16	20	4	4×500	2/中
Standard_A4m_v2	4	32	40	8	8×500	4/高
Standard_A8m_v2	8	64	80	16	16×500	8/高



表 5.1-8

型 号	CPU 核数	内存/GB	本地 SSD 磁盘/GB	最大数据磁盘数	数据磁盘最大吞吐量/IOPS	最大网卡数/网络带宽等级
Standard_D1_v2	1	3.5	50	2	2×500	1/中
Standard_D2_v2	2	7	100	4	4×500	2/高
Standard_D3_v2	4	14	200	8	8×500	4/高
Standard_D4_v2	8	28	400	16	16×500	8/高
Standard_D5_v2	16	56	800	32	32×500	8/极高
Standard_D11_v2	2	14	100	4	4×500	2/高
Standard_D12_v2	4	28	200	8	8×500	4/高
Standard_D13_v2	8	56	400	16	16×500	8/高
Standard_D14_v2	16	112	800	32	32×500	8/极高
Standard_D15_v2	20	140	1000	40	40×500	8/极高

表 5.1-9

型 号	CPU 核数	内存/GB	本地 SSD 磁盘/GB	最大数据磁盘数	最大网卡数/网络带宽等级
Standard_DS1_v2	1	3.5	7	2	1/中
Standard_DS2_v2	2	7	14	4	2/高
Standard_DS3_v2	4	14	28	8	4/高
Standard_DS4_v2	8	28	56	16	8/高
Standard_DS11_v2	2	14	28	4	2/高
Standard_DS12_v2	4	28	56	8	4/高
Standard_DS13_v2	8	56	112	16	8/高
Standard_DS14_v2	16	112	224	32	8/极高

## 5.2 磁盘和映像的使用

### 5.2.1 虚拟机附加磁盘

新建一台虚拟机在默认情况下只有一块操作系统磁盘，以及一块临时磁盘。

临时磁盘不是持久化的磁盘，其读写速度要略高于系统磁盘或者普通的附加磁盘，所以通常用于存放一些临时数据，进行 swap 分区或者虚拟内存的配置，或者存放一些安装程序来使用。

系统磁盘同样不适用于存放应用数据，原因是系统磁盘空间本身并不大，对于 Windows 虚拟机而言，默认的系统盘只有 127GB；对于 Linux 而言，只有 30GB。所以对于后期可能大量增长用户数据来说，系统磁盘的空间实在太小了。

为了拓展虚拟机的磁盘空间，可以通过向虚拟机附加新的数据磁盘来实现。不同的虚拟机型号最多可附加数据磁盘的数量是不同的，通常的计算公式是：

虚拟机最多可附加的磁盘数量=虚拟机 CPU 核数×2

例如标准 A2 虚拟机是 2 核，最多可以附加 4 块数据磁盘。

(1) Windows 虚拟机附加数据磁盘。

对于 Windows 虚拟机来说，要附加一块新的数据磁盘，首先需要登录到 Azure 管理界面中，选中要附加磁盘的虚拟机，切换到“仪表盘”选项，在下方选择“附加空磁盘”，如图 5.2-1 所示。



图 5.2-1

在弹出的界面中填写磁盘的 vhd 文件的存储位置（容器），磁盘的自定义名称（文件名），要附加的磁盘的大小以及缓存配置（磁盘大小可在 1~1023GB 之间取值，这个限制是由于磁盘使用的是页存储，页存储的最大空间只有 1023GB），如图 5.2-2 所示。

该图是一个名为“将空磁盘附加到虚拟机”的对话框。对话框顶部有一个关闭按钮（X）。内部包含以下配置项：1. “虚拟机名称”：一个文本输入框，下方有模糊的灰色背景。2. “存储位置”：一个文本输入框，下方有模糊的灰色背景。3. “文件名”：一个文本输入框，内容为“DiskName”。4. “大小(GB)”：一个数字输入框，内容为“1023”，右侧有一个清除按钮（X）。5. “主机缓存选项”：包含三个按钮：“无”（当前被选中）、“只读”和“读/写”，右侧有一个问号帮助图标。对话框右下角有一个确认按钮，带有一个对勾图标。

图 5.2-2

完成附加空磁盘的操作后，登录到虚拟机中，打开**磁盘管理**界面，会自动识别出这块刚刚添加的磁盘，如图 5.2-3 所示。

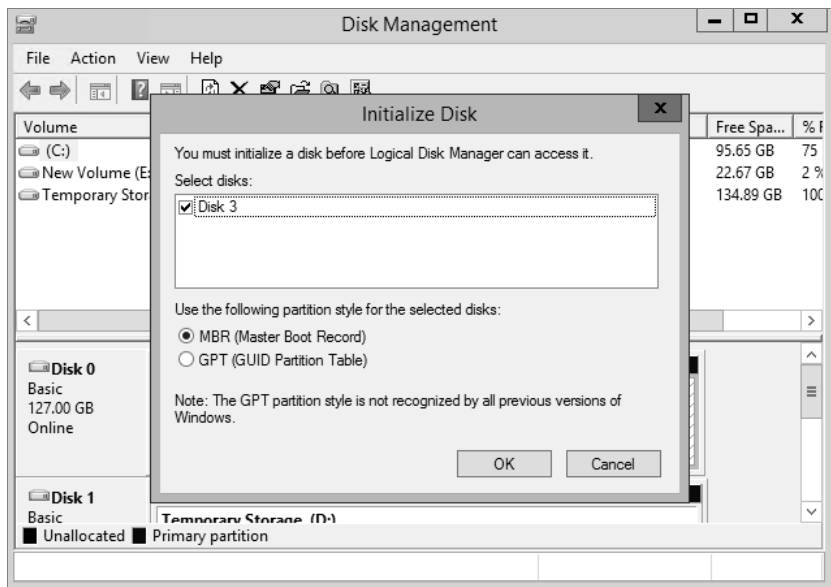


图 5.2-3

单击“OK”后，找到这块未格式化的磁盘，右键选择“New Simple Volume”（新建简单卷），如图 5.2-4 所示。

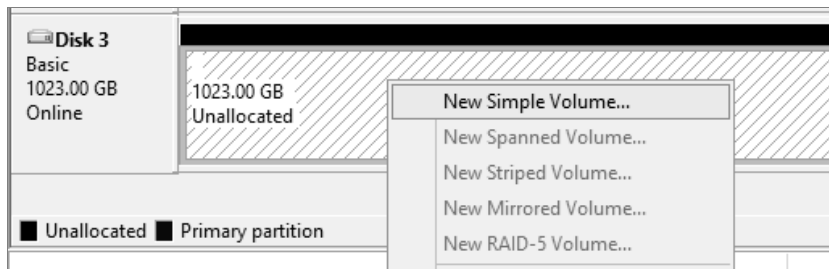


图 5.2-4

接着按照向导中的提示依次配置卷的大小、盘符、文件系统类型等，即可完成卷的创建。创建后会提示要对磁盘进行格式化，完成格式化后，就能够在系统中看到这块新的卷了，如图 5.2-5 所示。



图 5.2-5

## (2) Linux 虚拟机附加新磁盘。

要为 Linux 虚拟机附加新磁盘，同样需要在 Azure 管理界面先附加一块新的磁盘。附加完成后，登录虚拟机，使用 `fdisk -l` 查看是否能够正常识别到未分区的设备。

```
[root@Centos65 XXX]# fdisk -l
.....
Disk /dev/sdg: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

成功识别设备后，使用 fdisk 对设备进行分区：

```
[root@Centos65 XXX]# fdisk /dev/sdg
.....
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default 1305): 1305
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

写入分区后，使用 mkfs.ext4 为刚刚建立好的分区/dev/sdg1 建立 ext4 文件系统：

```
[root@Centos65 XXX]# mkfs.ext4 /dev/sdg1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2620595 blocks
131029 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

最后，创建一个挂载目录，将创建好文件系统的分区挂载到目录上：

```
[root@Centos65 XXX]# mkdir /mnt/sdg
[root@Centos65 XXX]# mount /dev/sdg1 /mnt/sdg
[root@Centos65 XXX]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	1007G	4.1G	952G	1%	/
tmpfs	3.5G	0	3.5G	0%	/dev/shm
/dev/sdb1	281G	191M	267G	1%	/mnt/resource
<b>/dev/sdg1</b>	<b>9.9G</b>	<b>151M</b>	<b>9.2G</b>	<b>2%</b>	<b>/mnt/sdg</b>

对于 Linux 虚拟机来说，如果希望每次系统启动后，分区都能够自动挂载，有两种方法可以实现。

一种是通过在/etc/fstab 文件中添加一个挂载项，系统启动后会读取这里面配置的挂载项进行依次挂载。需要注意的一点是，由于 Azure 虚拟机识别设备的顺序可能会与磁盘最初的挂载顺序不一致，所以不建议在/etc/fstab 文件中通过分区号（例如/dev/sdg1）进行挂载。因为如果识别顺序不同，或者有磁盘从虚拟机上被分离出来，分区号会发生变化，会直接导致虚拟机在启动的时候无法识别/etc/fstab 中写好的分区号而出现挂载失败，导致虚拟机出现启动失败的情况发生。为了避免这种情况，建议通过分区的 UUID 进行挂载，首先通过 blkid 查看分区的 UUID：

```
[root@Centos65 XXX]# blkid
.....
/dev/sdg1: UUID= " 60b08460-85c6-4b9c-9422-b0704b4b89d6 " TYPE= " ext4 "
```

在/etc/fstab 文件中添加一行挂载项：

```
UUID=60b08460-85c6-4b9c-9422-b0704b4b89d6 /mnt/sdg ext4 defaults 0 0
```

上面这种方法也存在风险，在这个分区文件系统出现异常的情况下，可能由于无法挂载异常分区引起虚拟机启动失败。要避免这个问题，可以通过在/etc/rc.local 中添加挂载命令来实现虚拟机开机后挂载分区，由于/etc/rc.local 的脚本是在虚拟机启动完成才执行，所以不会由于分区问题影响虚拟机的启动过程。在/etc/rc.local 中添加一行挂载脚本：

```
mount -U 60b08460-85c6-4b9c-9422-b0704b4b89d6 /mnt/sdg
```

在/etc/rc.local 中进行挂载要注意，如果系统中某些应用或者服务依赖于这里挂载的分区（例如数据库文件存储在这个分区中），那么这些应用或者服务一定要在分区挂载后启动。

## 5.2.2 捕获映像

在部署环境的过程中，如果遇到要部署多台相同服务器的情况，通常的解决方法是先部署一台虚拟机，然后在虚拟机内完成应用配置后，将这台虚拟机作为“模板”来批量“克隆”出其他虚拟机。这里提到的“模板”就是 Azure 的虚拟机映像。

Azure 经典模式中有两种不同的映像，一种是一般化（Generalized）的映像，另一种是特殊化（Specialized）的映像。

一般化的映像会去掉用户的配置信息（关于一般化具体的操作步骤以及说明，请参考下一节），所以使用一般化的映像创建虚拟机的时候，要进行一些额外的用户配置，例如需要配置虚拟机的登录用户名和密码等。

当使用一般化的映像创建虚拟机的时候，Azure 平台会再次对所创建的虚拟机进行**预配置**（Provision）。

使用一般化的映像创建虚拟机，如图 5.2-6 所示。



图 5.2-6

特殊化的映像包含了用户配置的完整信息，可以认为特殊化的映像与原虚拟机中的信息基本完全一致。使用特殊化的映像创建虚拟机的时候，不需要再进行额外的用户配置（虚拟网络、端口等配置除外，这些配置为 Azure 平台相关配置）。同时，Azure 平台不会使用特殊化映像创建的虚拟机再次进行预配置。

使用特殊化的映像创建虚拟机，如图 5.2-7 所示。



图 5.2-7

可以通过 Azure 管理界面将虚拟机捕获为映像，捕获映像的功能按钮在虚拟机仪表板正下方，如图 5.2-8 所示。



图 5.2-8

单击“捕获”按钮弹出“捕获虚拟机”对话框，如图 5.2-9 所示。



图 5.2-9

需要为映像指定一个名称便于记忆，同时可以为映像提供一个用于说明的标签。

在这个对话框的最下方有一个非常重要的选项“我已在虚拟机上运行“waagent-deprovision””（如果是 Windows 虚拟机捕获映像，则下面的选项为“我已在虚拟机上运行 Sysprep”），勾选这个选项表示已经对虚拟机系统进行了一般化操作。要将虚拟机捕获为一般化的映像，需要注意的是，将虚拟机捕获为一般化的映像后，这台虚拟机会被自动删除，如图 5.2-10 所示。

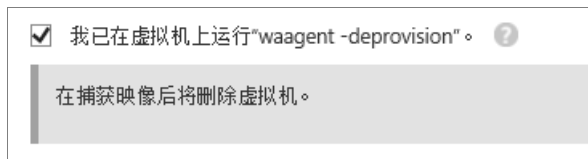


图 5.2-10

如果要捕获特殊化的映像，不需要对虚拟机系统进行一般化，直接在界面中捕获，不要勾选上述选项即可。

需要注意的是，虚拟机系统的状态一定要与映像的状态一致，即如果虚拟机系统进行了一般化处理，则必须将其捕获为一般化的映像；如果虚拟机未进行一般化处理，则必须将其捕获为特殊化的映像。如果不这样操作，则可能会出现使用捕获的映像创建虚拟机出

现**预配超时**的情况。例如出现下面这种情况。

有一台虚拟机未进行一般化，通过界面将这台虚拟机捕获为一般化的映像，在使用此映像创建新的虚拟机时，Azure 平台认为这台虚拟机的映像中未包含用户配置，会再次对这台虚拟机进行预配置。然而实际情况是，这个映像在捕获前实际未进行一般化操作，本身包含用户配置。在这种情况下，会导致平台对虚拟机的预配置操作与原有配置冲突，无法完成虚拟机的预配置，从而导致虚拟机状态变为预配超时。

除了用上面的方法捕获映像，也可以通过已有的包含操作系统的 vhd 文件来创建映像。这种创建方法也需要注意上面提到的情况，由于使用 vhd 文件创建映像时，只能创建一般化的映像，所以用来创建映像的 vhd 文件包含的操作系统必须是进行过一般化的系统。

### 5.2.3 虚拟机操作系统一般化

在捕获一般化映像前，首先需要对操作系统进行一般化操作。对于 Windows 虚拟机，使用 sysprep 工具来进行一般化；对于 Linux 虚拟机，则需要通过 waagent 来进行一般化。一般化后的虚拟机由于已经去掉了相关配置信息，所以无法继续使用。

#### （1）Windows 虚拟机一般化。

要针对 Windows 虚拟机进行一般化，首先需要登录虚拟机，在**运行**中输入“sysprep”，如图 5.2-11 所示。

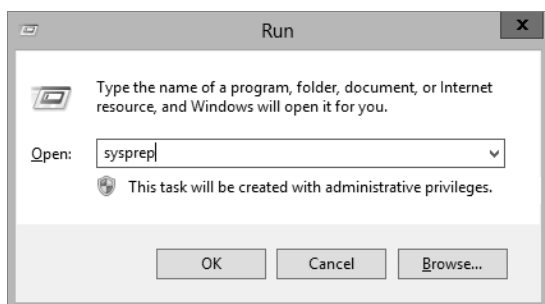


图 5.2-11

单击“OK”后打开 sysprep 工具所在目录，如图 5.2-12 所示。

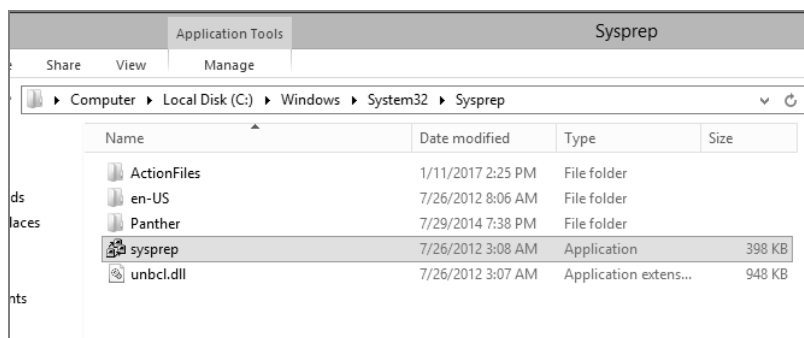


图 5.2-12



以管理员权限运行 sysprep.exe 工具，在**系统准备工具**的对话框中，选择“进入系统全新体验（OOBE）”，确保选中“一般化”的复选框，同时**关机选项**选择“关机”，如图 5.2-13 所示。

单击“OK”确认后，系统开始自动进行一般化的操作，操作完成后，虚拟机会自动关机，如图 5.2-14 所示。

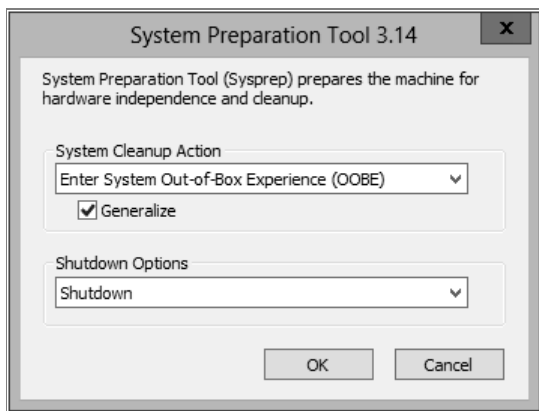


图 5.2-13

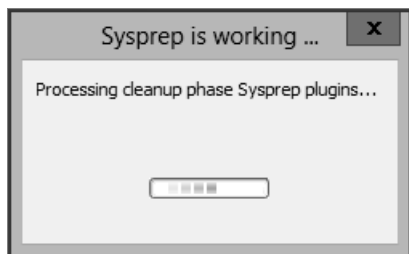


图 5.2-14

关于 sysprep 的工作原理，可以参考链接：[https://technet.microsoft.com/zh-cn/library/dd744512\(v=ws.10\).aspx](https://technet.microsoft.com/zh-cn/library/dd744512(v=ws.10).aspx)。

## （2）Linux 虚拟机一般化。

要对 Linux 虚拟机进行一般化，需要使用 Linux 虚拟机代理（waagent）来操作。通常 waagent 都会在创建虚拟机的同时安装配置，如果未安装 waagent，可以参考后面相关章节进行安装。

在安装 waagent 的虚拟机中执行下面的命令进行一般化：

```
waagent -deprovision
```

waagent 对虚拟机进行一般化主要包含以下几个步骤：

- a. 清除 SSH 主机密钥。
- b. 清除/etc/resolv.conf 中的 nameserver 配置。
- c. 清除/etc/shadow 中的根密码。
- d. 清除缓存的 DHCP 客户端租用。
- e. 将主机名重置为 localhost.localdomain。

需要注意的是，建议使用上面的命令进行一般化，不建议使用下面的命令进行一般化，下面的命令会删除默认管理员用户 home 下的用户目录（/home/defaultadmin）下的数据：

```
waagent -deprovision user
```

## 5.2.4 自定义映像

对于希望将本地预先配置好的虚拟机作为模板上传到 Azure 中的用户，Azure 也提供

了用户自定义映像的选择。在将本地虚拟机的操作系统磁盘上传到 Azure 之前，首先要了解下面的一些常见的注意事项。

#### （1）磁盘格式。

Azure 不支持 VHDX 格式的磁盘，仅支持**固定大小的 VHD**，VHD 大小必须是 1MB 的倍数。对于 VHDX 磁盘，可以使用 Hyper-V 管理器或者 Convert-VHD 命令将 VHDX 转换为 VHD。如果本地使用的是 VMWare 创建的虚拟机，则可以使用 Microsoft 虚拟机转换器将 VMDK 转换为 VHD。

使用 Hyper-V 管理器进行磁盘格式转换可以参考下面的步骤。

- a. 打开 Hyper-V 管理器，在左侧选择“本地计算机”。在本地计算机上面的菜单中，单击“操作”—“编辑磁盘”。
- b. 在“查找虚拟硬盘”屏幕上，浏览并选择要转换的虚拟磁盘。
- c. 在“选择操作”屏幕上，依次选择“转换”和“下一步”。
- d. 如果需要从 VHDX 进行转换，应选择“VHD”，然后单击“下一步”。
- e. 如果需要从动态扩展磁盘进行转换，应选择“固定大小”，然后单击“下一步”。
- f. 浏览并选择要保存新 VHD 文件的路径。
- g. 单击“完成”以关闭。

使用下面的 Convert-VHD 命令也可以完成磁盘格式转换，并将磁盘从动态扩展转换为固定大小的磁盘：

```
Convert-VHD -Path D:\XXXX.vhdx -DestinationPath D:\XXXX.vhd -VHDType Fixed
```

使用 Microsoft 虚拟机转换器将 VMDK 转换为 VHD：

转换器下载地址：<http://www.microsoft.com/en-us/download/details.aspx?id=42497>

下载后使用默认选项安装即可，假定安装目录为“C:\Program Files”，使用下面的 Powershell 命令导入转换器的模块：

```
Import-Module 'C:\Program Files\Microsoft Virtual Machine Converter\MvmcCmdlet.psd1'
```

导入成功后，使用 Get-Module 查看 MvmcCmdlet 模块是否已经成功导入：

```
PS C:\Users\DanielHX> Get-Module | select Name, Version
```

Name	Version
Microsoft.PowerShell.Management	3.1.0.0
Microsoft.PowerShell.Utility	3.1.0.0
<b>MvmcCmdlet</b>	<b>2.0</b>
PSReadline	1.1

成功导入模块后，就可以使用下面的命令进行格式转换了：

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath D:\XXXX.vmdk -VhdType FixedHardDisk -VhdFormat Vhd -DestinationLiteralPath D:\XXXX.vhd
```

### (2) 虚拟机代次。

Azure 仅支持 Hyper-V 创建的一代虚拟机，虚拟机在创建后不能从二代调整为一代。关于创建一代或二代虚拟机的详细说明，可以参考链接：<https://technet.microsoft.com/windows-server-docs/compute/hyper-v/plan/should-i-create-a-generation-1-or-2-virtual-machine-in-hyper-v>。

### (3) 虚拟机配置时的注意事项。

Linux 虚拟机在安装系统时，建议使用标准分区而非 LVM 分区（通常 LVM 是安装时的默认选项），LVM 和 RAID 建议仅在数据磁盘中使用。

由于低于 2.6.37 的 Linux 内核版本中的 bug，大型号的 VM 不支持 NUMA。此问题主要影响 Red Hat 2.6.32 及较早的内核版本。手动安装的 Azure Linux 代理（waagent）将自动在 Linux 内核的 GRUB 配置中禁用 NUMA。

不要在操作系统磁盘上配置交换分区。

将网卡配置为 DHCP 模式而非静态地址，删除任何与本地环境有关的自定义路由表。

配置防火墙规则，放行关键服务（例如 RDP 和 SSH 服务）。

有关为上传准备自定义映像的具体操作步骤，可以进一步参考以下 Azure 官网指南。

准备要上传到 Azure 的 Windows 虚拟机：<https://www.azure.cn/documentation/articles/virtual-machines-windows-prepare-for-upload-vhd-image/>

准备要上传到 Azure 的 CentOS 虚拟机：<https://www.azure.cn/documentation/articles/virtual-machines-linux-create-upload-centos/>

准备要上传到 Azure 的 Ubuntu 虚拟机：<https://www.azure.cn/documentation/articles/virtual-machines-linux-create-upload-ubuntu/>

准备要上传到 Azure 的 Debian 虚拟机：<https://www.azure.cn/documentation/articles/virtual-machines-linux-debian-create-upload-vhd/>

准备要上传到 Azure 的 RedHat 虚拟机：<https://www.azure.cn/documentation/articles/virtual-machines-linux-redhat-create-upload-vhd/>

## 5.3 Linux 虚拟机图形化配置

### 5.3.1 CentOS (OpenLogic) 图形化配置

本文主要介绍如何在 Azure 上为 CentOS 的虚拟机安装图形界面。实验环境为 Azure 虚拟机，系统版本为 CentOS 6.8，如果是其他版本的 CentOS，可能在配置方法和步骤上会有不同，因此其他版本不保证一定可行。

#### 准备安装环境

本文以 CentOS 6.8 系统版本为例。

图形化配置操作如下：

首先登录 CentOS 虚拟机，并且切换 root：

```
[root@CentOS68 ~]# yum grouplist
```

```
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Group Process
Determining fastest mirrors
base | 3.7 kB    00:00
...
    Zulu Support [zu]
Done
[root@sunCentOS68 ~]#
```

### #安装 “Desktop” 环境

```
[root@CentOS68 ~]# yum groupinstall -y 'X Window System'

[root@CentOS68 ~]# yum groupinstall -y "Desktop" --skip-broken
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Group Process
Loading mirror speeds from cached hostfile
Package nautilus-2.28.4-25.el6.x86_64 already installed and latest version
...
Complete!
[root@sunCentOS68 ~]#
```

### 安装 VNC Server+配置

#### #安装 tigervnc

```
[root@jcentos68 ~]# yum install -y tigervnc*
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
...
Complete!
```

#### #安装 libXfont

```
[root@centos68 ~]# yum install -y libXfont*
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Package libXfont-1.5.1-2.el6.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
...
Complete!
[root@sunjcentos68 ~]#
```

#### #安装 pixman、xterm、xorg-x11-twm

```
[root@sunjcentos68 ~]# yum install -y pixman xterm xorg-x11-twm
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Package pixman-0.32.8-1.el6.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
...

Complete!
[root@sunjcentos68 ~]#
```

**#vi /etc/sysconfig/vncservers**

插入以下两行内容:

**VNCSERVERS= " 1:root "**

**VNCSERVERARGS[1]= " -geometry 800×600 "**

```
# The VNCSERVERS variable is a list of display:user pairs.
#
# Uncomment the lines below to start a VNC server on display :2
# as my 'myusername' (adjust this to your own). You will also
# need to set a VNC password; run 'man vncpasswd' to see how
# to do that.
#
# DO NOT RUN THIS SERVICE if your local area network is
# untrusted! For a secure way of using VNC, see this URL:
# https://access.redhat.com/knowledge/solutions/7027

# Use " -nolisten tcp " to prevent X connections to your VNC server via TCP.

# Use " -localhost " to prevent remote VNC clients connecting except when
# doing so through a secure tunnel. See the " -via " option in the
# `man vncviewer' manual page.

# VNCSERVERS= " 2:myusername "
# VNCSERVERARGS[2]= " -geometry 800×600 -nolisten tcp -localhost "
VNCSERVERS= " 1:root "
VNCSERVERARGS[1]= " -geometry 800×600 "

~
~
-- INSERT --
```

**设置 VNC 密码**

```
[root@sunjcentos68 ~]# vncpasswd
Password:
Verify:
[root@sunjcentos68 ~]#
```

## 设置 CentOS 分辨率

打开/boot/grub/grub.conf，在文件最后新添加一行参数 **vga=789**。

**vga=789** 表示 800×600，16M 色彩

```
[root@sunjcentos68 ~]# vi /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#         all kernel and initrd paths are relative to /, eg.
#         root (hd0,0)
#         kernel /boot/vmlinuz-version ro root=/dev/sda1
#         initrd /boot/initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS 6 (2.6.32-642.13.1.el6.x86_64)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.32-642.13.1.el6.x86_64 ro root=UUID=
d77e4362-c9d7-4972-ae7c-f5b2cela43a1 rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us
LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 console=ttyS0,115200n8
earlyprintk=ttyS0,115200 rootdelay=300 rd_NO_LVM rd_NO_DM
    initrd /boot/initramfs-2.6.32-642.13.1.el6.x86_64.img
vga=789
~
~
-- INSERT --
```

## 启动 VNC 服务（注意“:1”前有空格）

```
[root@sunjcentos68 ~]# vncserver :1
xauth: file /root/.Xauthority does not exist

New 'sunjcentos68:1 (root)' desktop is sunjcentos68:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/sunjcentos68:1.log

[root@sunjcentos68 ~]#
```

## NSG 配置

ARM 默认有 NSG 因此需要添加允许 VNC TCP 5901 终结点，如图 5.3-1 所示。

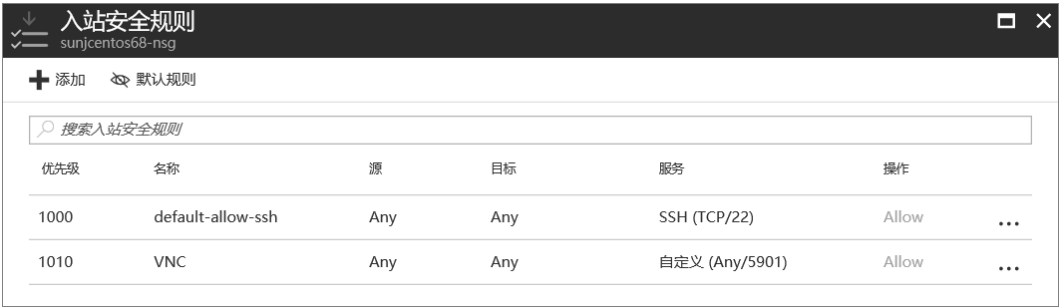


图 5.3-1

下载并安装启动 VNC Viewer，如图 5.3-2 所示。

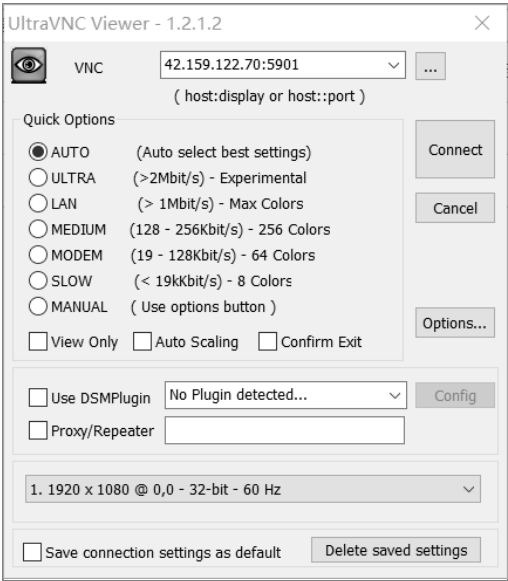


图 5.3-2

输入密码，如图 5.3-3 所示。

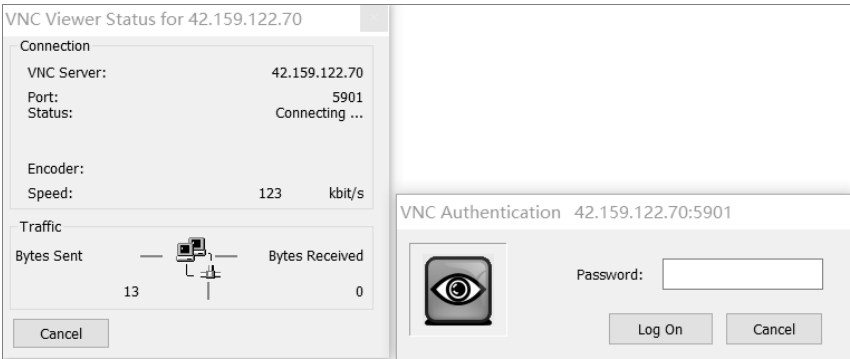


图 5.3-3

登录成功，如图 5.3-4 所示。

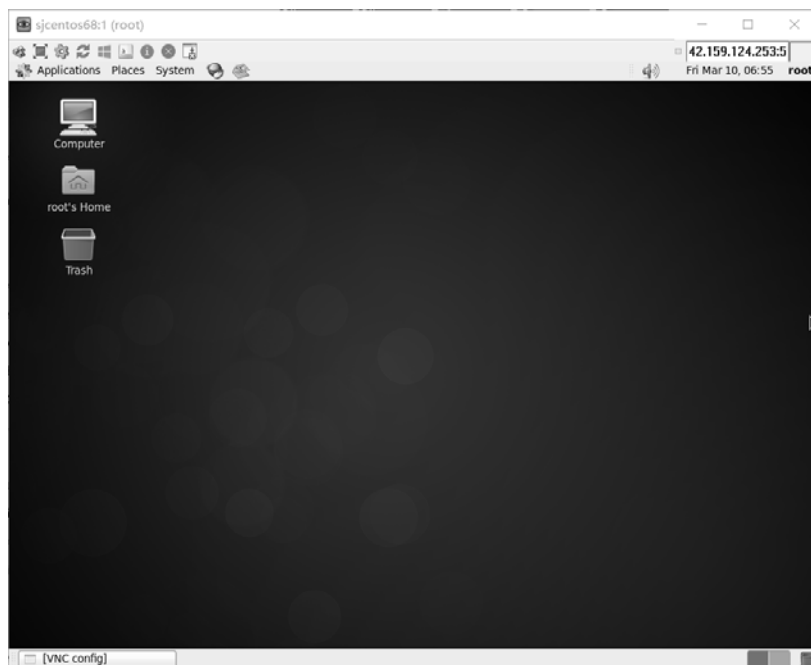


图 5.3-4

### 5.3.2 Ubuntu 图形化配置

本节主要介绍如何在 Azure 上为 Ubuntu 的虚拟机安装图形界面。实验环境为 Azure 虚拟机，系统版本为 Ubuntu 12.04，如果是其他版本的 Ubuntu，可能在配置方法和步骤上会有不同，因此其他版本不保证一定可行。

#### 准备安装环境

本文以 Ubuntu 12.04 系统版本为例。

图形化配置操作如下：

首先在新版门户上创建一台 Ubuntu 12.04 的虚拟机（这里用的是 ARM 下的映像）。

登录 Ubuntu 虚拟机。

创建完成后使用 XShell 登录虚拟机，依次执行下面的命令。

安装 xfce4：

```
root@sjubuntu12:~#apt-get install -y xfce4
...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Processing triggers for libgdk-pixbuf2.0-0 ...
Processing triggers for initramfs-tools ...
update-initramfs: Generating /boot/initrd.img-3.13.0-112-generic
root@sjubuntu12:~#
```



安装 xrdp 组件和 vnc 服务器:

```
root@sjubuntu12:~# apt-get install xrdp vnc4server
Reading package lists... Done
Building dependency tree
...
* Starting Remote Desktop Protocol server [ OK ]
root@sjubuntu12:~#
```

生成一个默认的配置文件:

```
root@sjubuntu12:~# echo "xfce4-session" > ~/.xsession

重启 xrdp 服务:
root@sjubuntu12:~# service xrdp restart
* Stopping RDP Session manager [ OK ]
* Starting Remote Desktop Protocol server [ OK ]
root@sjubuntu12:~#
```

ARM 下的虚拟机如果配置了 NSG，则需要注意 NSG 3389 的入站默认规则是否存在，如果是经典虚拟机，则需要为虚拟机添加 remote desktop（3389）终结点，如图 5.3-5 所示。

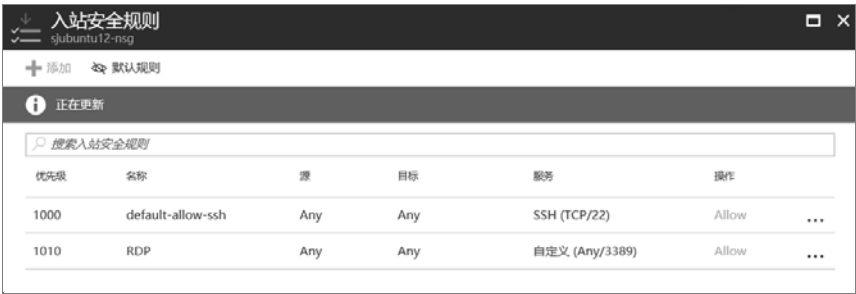


图 5.3-5

NSG 完成后，我们可以用 mstsc 直接与 Public IP 进行连接，如图 5.3-6 所示。



图 5.3-6

登录后输入用户名密码，如图 5.3-7 所示。

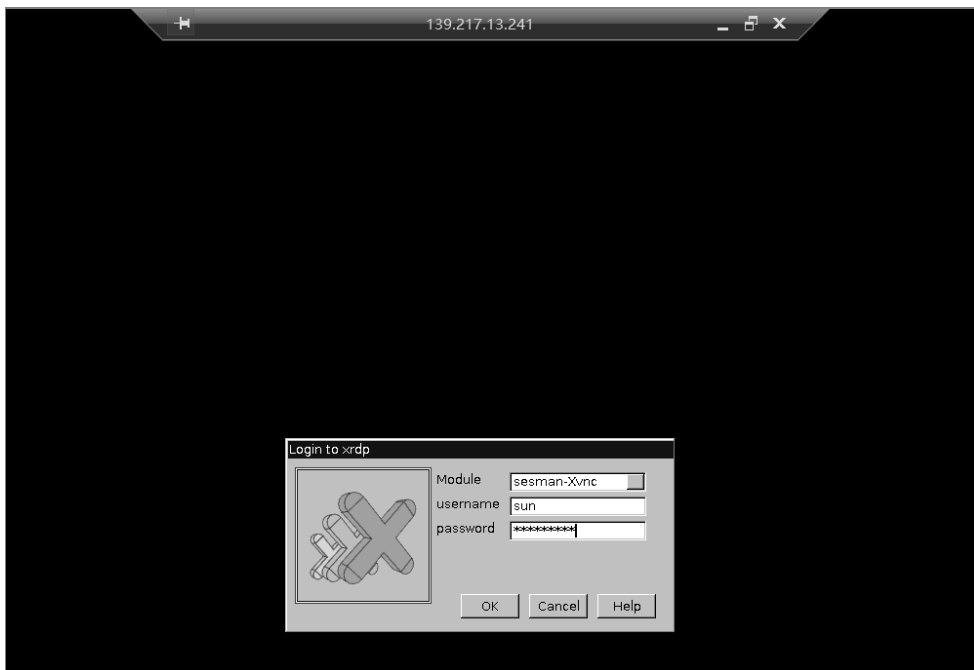


图 5.3-7

然后正常情况下即可进入图形界面，如图 5.3-8 所示。



图 5.3-8

### 5.3.3 SUSE 图形化配置

本节主要介绍如何在 Azure 上为 SUSE Linux Enterprise Server 12 SP2 x86\_64 的虚拟机安装图形界面。实验环境为 Azure 虚拟机，系统版本为 SUSE Linux Enterprise Server 12 SP2 x86\_64，如果是其他版本的 SUSE，可能在配置方法和步骤上会有不同，因此其他版本不保证一定可行。

## 准备安装环境

在配置之前，首先要进行下面的准备工作：

在 Azure 上创建 SUSE Linux Enterprise Server 12 SP2 x86\_64 的虚拟机，如图 5.3-9 所示。

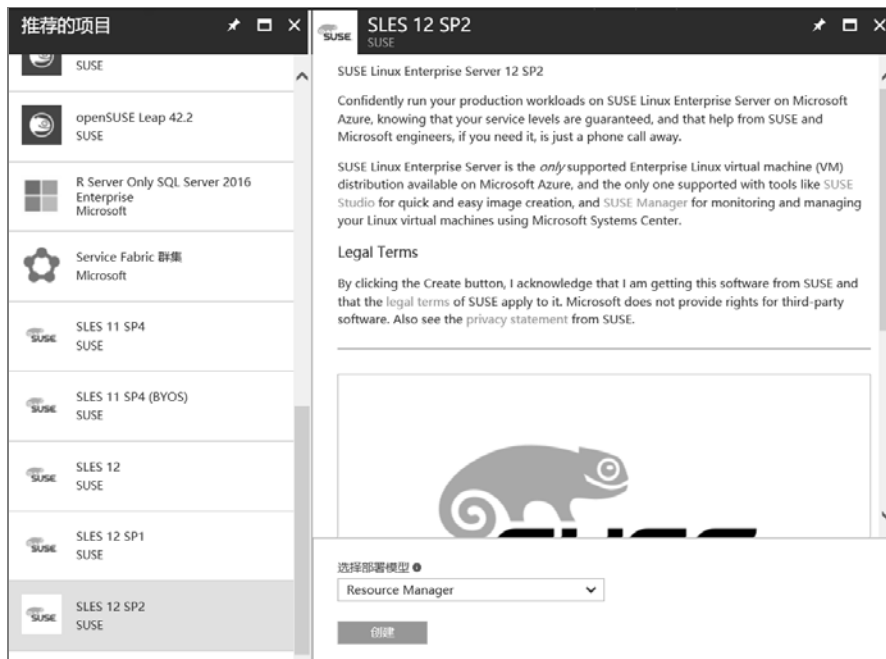


图 5.3-9

图形化配置操作如下。

首先登录 SUSE Linux 虚拟机并安装软件，确保以“root”身份登录。

### 安装 gnome-session

```
sjsuse-12-sp2:~ # zypper install gnome-session
Refreshing service 'SMT-http_smt-azure_susecloud_net'.
...
Overall download size: 183.5 MiB. Already cached: 0 B. After the operation,
additional 649.7 MiB will be used.
Continue? [y/n/? shows all options] (y): y
...
Running: cantarell-fonts-0.0.24-3.1-reconfigure-fonts (cantarell-fonts,
/var/adm/update-scripts) .....[done]
sjsuse-12-sp2:~ #
```

### 安装 gnome-basic

```
sjsuse-12-sp2:~ # zypper install -t pattern gnome-basic
Refreshing service 'SMT-http_smt-azure_susecloud_net'.
Refreshing service 'cloud_update'.
Loading repository data...
...
```

```

317 new packages to install.
Overall download size: 215.2 MiB. Already cached: 0 B. After the operation,
additional 650.6 MiB will be used.
Continue? [y/n/? shows all options] (y): y
Running:          xterm-308-3.8-reconfigure-fonts                (xterm,
/var/adm/update-scripts) .....
.....
...[done]
sjsuse-12-sp2:~ #

```

### 下载 xrdp-0.9.0~git.1456906198.f422461-7.1.x86\_64.rpm 安装包

```

sjsuse-12-sp2:~ # zypper in http://download.opensuse.org/repositories/
X11:/RemoteDesktop/opensUSE_13.2/x86_64/xrdp-0.9.0~git.1456906198.f422461-7
.1.x86_64.rpm
Refreshing service 'SMT-http_smt-azure_susecloud_net'.
Refreshing service 'cloud_update'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  xrdp

The following package is not supported by its vendor:
  xrdp

1 new package to install.
Overall download size: 352.2 KiB. Already cached: 0 B. After the operation,
additional 1.9 MiB will be used.
Continue? [y/n/? shows all options] (y): y
Retrieving      package      xrdp-0.9.0~git.1456906198.f422461-7.1.x86_64
(1/1), 352.2 KiB (  1.9 MiB unpacked)
xrdp-0.9.0~git.1456906198.f422461-7.1.x86_64.rpm:
  Header V3 DSA/SHA1 Signature, key ID 0f2672c8: NOKEY
  V3 DSA/SHA1 Signature, key ID 0f2672c8: NOKEY

xrdp-0.9.0~git.1456906198.f422461-7.1.x86_64 (Plain RPM files cache):
Signature verification failed [4-Signatures public key is not available]
Abort, retry, ignore? [a/r/i] (a): i
Checking                for                file
conflicts: .....
.....
.....[done]
(1/1)                                Installing:
xrdp-0.9.0~git.1456906198.f422461-7.1.x86_64 .....
.....
.....[done]
Additional rpm output:

```

```
warning:
/var/cache/zypp/packages/_tmpRPMcache_/xrdp-0.9.0~git.1456906198.f422461-7.
1.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 0f2672c8: NOKEY
Updating /etc/sysconfig/xrdp...

Generating 512 bit rsa key...

ssl_gen_key_xrdp1 ok

saving to /etc/xrdp/rsakeys.ini

sjsuse-12-sp2:~ #
```

### 安装 tigervnc xorg-x11-Xvnc xterm

```
sjsuse-12-sp2:~ # zypper install tigervnc xorg-x11-Xvnc xterm
Refreshing service 'SMT-http_smt-azure_susecloud_net'.
Refreshing service 'cloud_update'.
Loading repository data...
Reading installed packages...
'tigervnc' is already installed.
No update candidate for 'tigervnc-1.6.0-16.4.x86_64'. The highest available
version is already installed.
'xterm' is already installed.
No update candidate for 'xterm-308-3.8.x86_64'. The highest available version
is already installed.
'xorg-x11-Xvnc' is already installed.
No update candidate for 'xorg-x11-Xvnc-1.6.0-16.4.x86_64'. The highest
available version is already installed.
Resolving package dependencies...

Nothing to do.
sjsuse-12-sp2:~ #
```

### 开启 xrdp

```
#systemctl start xrdp
```

```
sjsuse-12-sp2:~ # systemctl start xrdp
```

### 启动 xrdp

```
#systemctl enable xrdp
```

```
sjsuse-12-sp2:~ # systemctl enable xrdp
Created symlink from /etc/systemd/system/multi-user.target.wants/
xrdp.service to /usr/lib/systemd/system/xrdp.service.
sjsuse-12-sp2:~ #
```

### VNC 设置登录密码和服务配置

设置 VNC 登录密码

```
sjsuse-12-sp2:~ # vncpasswd
Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:
sjsuse-12-sp2:~ #
```

**start with display number '1', 屏幕分辨率'1024×768', 颜色深度'24'**

```
sjsuse-12-sp2:~ # vncserver :1 -geometry 1024x768 -depth 24
xauth: file /root/.Xauthority does not exist

New 'sjsuse-12-sp2:1 (root)' desktop is sjsuse-12-sp2:1

Creating default startup script /root/.vnc/xstartup
Creating default config /root/.vnc/config
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/sjsuse-12-sp2:1.log
sjsuse-12-sp2:~ #
```

因为 ARM 环境默认没有开放 5901，需要 NSG 网络安全组，手动添加 5901 端口，如图 5.3-10 所示。



图 5.3-10

在客户端计算机上安装 VNC 客户端，此示例在 Windows 10 上。

从以下网站下载安装 UltraVNC。

<http://www.uvnc.com/downloads/ultravnc.html>

安装 UltraVNC 后，单击“UltraVNC Viewer”运行，出现如图 5.3-11 所示的界面。输入“服务器的主机名或 IP 地址: 端口号”，然后单击“连接”按钮（VNC 默认端口是 5901）。

输入密码，如图 5.3-12 所示。

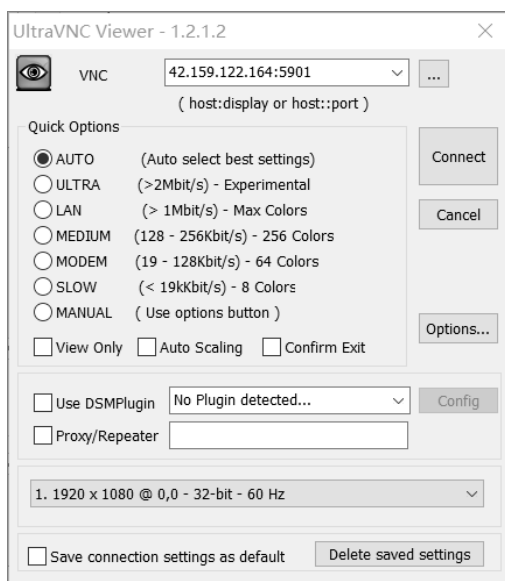


图 5.3-11



图 5.3-12

登录成功，如图 5.3-13 所示。

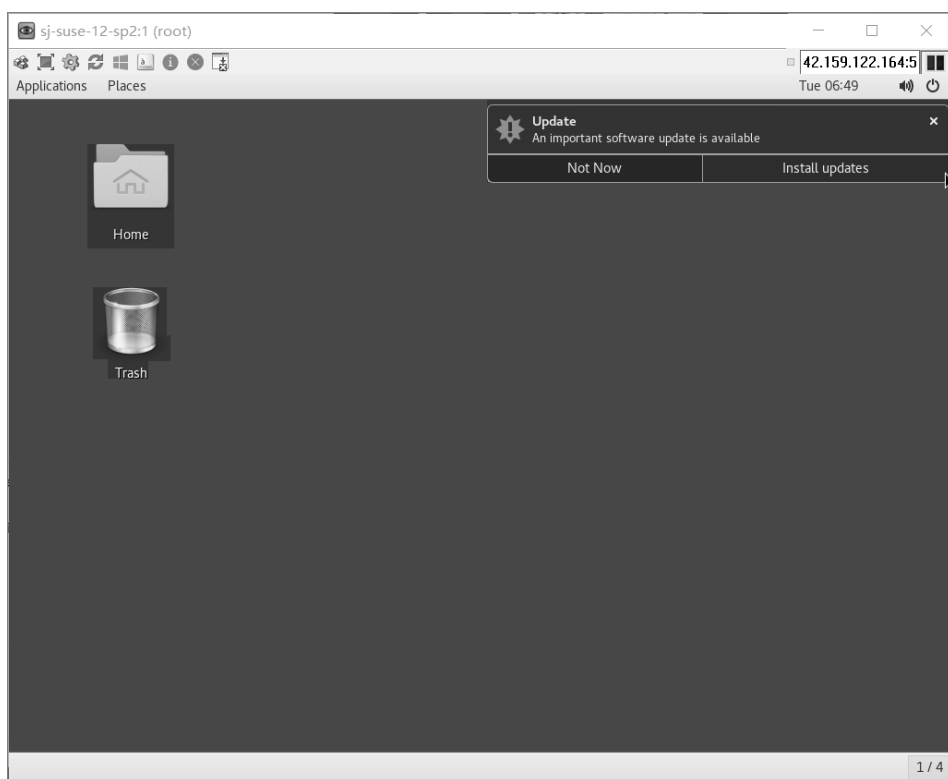


图 5.3-13

## 5.4 虚拟机扩展的介绍和使用

### 5.4.1 VM Agent 简介

VM Agent 是部署在虚拟机内的一个进程，用于协助管理、配置和加速虚拟机。在使用官方映像创建虚拟机的最后一个步骤中有配置 VM Agent 的选项。

对于 Windows 虚拟机而言，常见的一些 VM Agent 的应用，例如在背景中看到的虚拟机信息，就是由 VM Agent 安装的扩展 BGInfo 配置的，如图 5.4-1 所示。

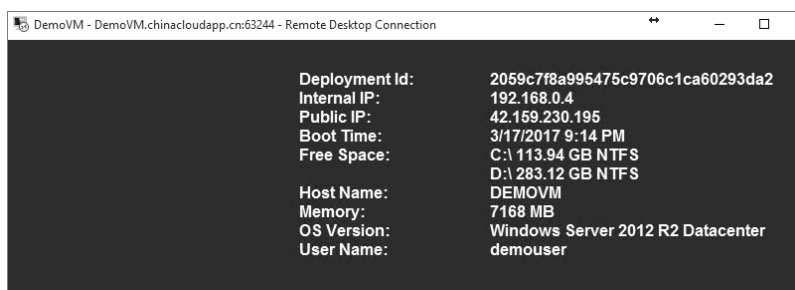


图 5.4-1

还可以通过 VM Agent 安装扩展 VMAccess 来重置用户名密码，通过 VM Agent 进行用户自定义脚本的部署，在部署虚拟机规模集的时候批量推送预配脚本等。除了配置扩展外，VM Agent 本身还会负责一些数据采集，心跳检测，甚至在备份功能中也起到关键作用。在虚拟机的预配置阶段，VM Agent 会进行一些配置，例如用户创建、SSH 服务配置、主机名配置、DNS 设置，等等。

Linux 虚拟机中的 VM Agent 也有类似作用。

可以通过 Azure Powershell 命令查看全部可用的扩展程序：

```
PS C:\Users\XXX> Get-AzureVMAvailableExtension | Select ExtensionName,
Publisher
```

ExtensionName	Publisher
AsiaInfoDSA	AsiaInfo.DeepSecurity
IaaS.Diagnostics	Microsoft.Azure.Diagnostics
CustomScript	Microsoft.Azure.Extensions
DockerExtension	Microsoft.Azure.Extensions
VMSnapshot	Microsoft.Azure.RecoveryServices
VMSnapshotLinux	Microsoft.Azure.RecoveryServices
IaaS.Antimalware	Microsoft.Azure.Security
AzureCATExtensionHandler	Microsoft.AzureCAT.AzureEnhancedMonitoring
BGInfo	Microsoft.Compute
CustomScriptExtension	Microsoft.Compute
JsonAddDomainExtension	Microsoft.Compute
VMAccessAgent	Microsoft.Compute



LinuxNodeAgent	Microsoft.HpcPack
CustomScriptForLinux	Microsoft.OSTCExtensions
DSCForLinux	Microsoft.OSTCExtensions
LinuxDiagnostic	Microsoft.OSTCExtensions
OSPatchingForLinux	Microsoft.OSTCExtensions
VMAccessForLinux	Microsoft.OSTCExtensions
DSC	Microsoft.Powershell
DSC.Edp	Microsoft.Powershell
DSC	Microsoft.Powershell.Test
SqlIaaSAgent	Microsoft.SqlServer.Management
VS14CTPDebugger	Microsoft.VisualStudio.Azure.RemoteDebug
VS2012Debugger	Microsoft.VisualStudio.Azure.RemoteDebug
VS2013Debugger	Microsoft.VisualStudio.Azure.RemoteDebug
VSRemoteDebugger	Microsoft.VisualStudio.Azure.RemoteDebug
WebDeployForVSDevTest	Microsoft.VisualStudio.WindowsAzure.DevTest
DockerExtension	MSOpenTech.Extensions

### 5.4.2 VM Agent 安装

如果使用平台映像创建虚拟机，默认在步骤四会有选项确认“安装 VM 代理”，默认该选项是勾选状态。勾选了该选项后，虚拟机在完成创建的时候 VM Agent 会自动部署在虚拟机中。

对于没有勾选该选项的虚拟机（如果是使用平台映像部署 Linux 虚拟机，VM Agent 选项是必选的，无法取消勾选），或者是用户自定义上传的映像，要使用 VM Agent 的功能，就需要手动安装 VM Agent。

#### （1）Windows 虚拟机安装 VM Agent。

首先通过链接下载 VM Agent 的 msd 安装程序：<http://go.microsoft.com/fwlink/?LinkID=394789&clcid=0x409>。

下载后，使用管理员权限安装，如图 5.4-2 所示。

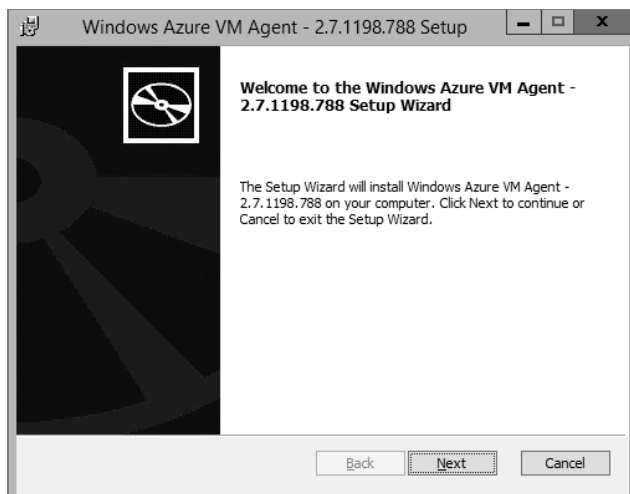


图 5.4-2

安装完成后可以在任务管理器中看到 WaAppAgent.exe 和 WindowsAzureGuestAgent.exe 两个进程，如图 5.4-3 所示。

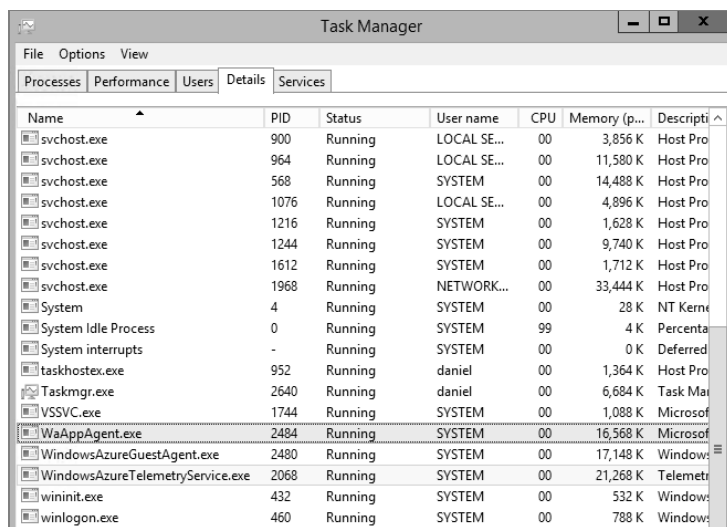


图 5.4-3

到此还没有安装完成，虽然虚拟机内已经安装好了 VM Agent，但是在 Azure 平台上，还没有标记这台虚拟机已经安装了 VM Agent。所以还需要通过 Azure Powershell 来将此虚拟机标记为已安装 VM Agent 的状态。

通过下面两条命令查看虚拟机的 WA Agent 标记状态：

```
PS C:\Users\XXX> $vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoVM
PS C:\Users\XXX> $vm.VM.ProvisionGuestAgent
False
```

从命令输出看到，虚拟机的 ProvisionGuestAgent 状态还是 False，通过下面的命令将其更新为 True：

```
PS C:\Users\XXX> $vm.VM.ProvisionGuestAgent = $true
PS C:\Users\XXX> $vm | Update-AzureVM
```

到此，VM Agent 就已经安装完成了。

## (2) Linux 虚拟机安装 VM Agent。

Linux 上安装 VM Agent，可以直接通过 yum, apt-get 等软件包管理工具进行安装，这里以 CentOS 为例，通过下面的命令安装 VM Agent（注意这里需要区分 WALinuxAgent 的大小写）：

```
[root@XXX ~]# yum install WALinuxAgent
```

安装完成后，使用下面的命令查看 VM Agent 是否安装成功：

```
[root@XXX ~]# service waagent status
waagent is stopped
```

可以看到，默认 VM Agent 没有启动，通过下面的命令启动 VM Agent 服务：

```
[root@XXX ~]# service waagent start
```

查看 VM Agent 是否开机自动启动：

```
[root@XXX ~]# runlevel
N 3
[root@XXX ~]# chkconfig
.....
waagent          0:off   1:off   2:on    3:on    4:on    5:on    6:off
.....
```

可以看到在当前的运行级别上，VM Agent 是随开机启动的。

到这里，已经在虚拟机内完成了 VM Agent 的安装和配置，下一步就需要使用 Azure Powershell 将此虚拟机标记为已安装 VM Agent 的状态，具体的操作和前面对 Windows 虚拟机的操作相同。

### 5.4.3 重置密码与 SSH 配置

利用 VM Agent 在虚拟机中安装扩展，可以为虚拟机重置密码，还可以为虚拟机重置 SSH 配置。

#### 1. Windows 虚拟机重置密码

针对 Windows 虚拟机，可以使用下面的 Azure Powershell 命令进行密码重置：

```
PS C:\Users\XXX> $extension = "VMAccessAgent "
PS C:\Users\XXX> $publisher = "Microsoft.Compute "
PS C:\Users\XXX> $ver = "2.*"
PS C:\Users\XXX> $publicConf = '{ "UserName" : "DemoUser" }'
PS C:\Users\XXX> $privateConf = '{ "Password" : "Passw0rd!" }'
PS C:\Users\XXX> $vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoVM
PS C:\Users\XXX> Set-AzureVMExtension -VM $vm -ExtensionName $extension
-Publisher $publisher -Version $ver -Public
Configuration $publicConf -PrivateConfiguration $privateConf |
Update-AzureVM
```

在上面的命令中版本号我们指定 2.\* 的原因是，目前扩展 VMAccessAgent 的大版本是 2，通过制定小版本\*可以让 VM Agent 安装最新版本的 VMAccessAgent。

重置成功后，可以在虚拟机的仪表板中查看扩展的安装状态，如图 5.4-4 所示。

扩展			
名称	版本	状态	消息
Microsoft.Compute.VMAcce...	2.3	Success	Successfully updated build-in Admin account and enable...

图 5.4-4

可以看到扩展已经安装成功，版本是 2.3，从消息中看到已经成功为管理员用户重置了密码的说明。

这里有两点需要说明：

(1) 如果用来重置的用户名与原用户名不同，重置成功后，会将原来的管理员用户替换为新的重置用户，例如上面的例子中，指定的重置用户名为 **DemoUser**，重置成功后，原管理员用户就变成了 **DemoUser**，密码也变为了新密码。

(2) 由于 **VM Agent** 需要使用管理员用户权限执行操作，所以在管理员被禁用的情况下，或者管理员密码过期的情况下，是无法通过 **VM Agent** 重置密码的。

## 2. Linux 虚拟机重置密码

Linux 虚拟机重置密码的命令与 Windows 类似，也是使用 Azure Powershell 调用 **VM Agent** 为虚拟机安装重置密码的扩展工具，命令如下：

```
$extension = "VMAccessForLinux"
$publisher = "Microsoft.OSTCEExtensions"
$ver = "1.*"
$privateConf = '{ "username": "DemoUser", "password": "Passw0rd!" }'
$vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoLinuxVM
Set-AzureVMExtension -VM $vm -ExtensionName $extension -Publisher $publisher
-Version $ver -PrivateConfiguration $privateConf | Update-AzureVM
```

重置完成后，也可以在虚拟机仪表板中找到安装的扩展，如图 5.4-5 所示。

扩展			
名称	版本	状态	消息
Microsoft.OSTCEExtensions.V...	1.4.5.0	Success	Plugin enabled

图 5.4-5

对于 Linux 虚拟机而言，如果指定的用户名与原用户名不同，并不会将原管理员用户清除。

## 3. Linux 虚拟机重置 SSH 配置

重置 SSH 配置文件也是通过 **VMAccessForLinux** 扩展，不同的是 **privateConfiguration** 参数中传入的 json 参数，脚本如下：

```
$extension = "VMAccessForLinux"
$publisher = "Microsoft.OSTCEExtensions"
$ver = "1.*"
$privateConf = '{ "reset_ssh": "True" }'
$vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoLinuxVM
```

```
Set-AzureVMExtension -VM $vm -ExtensionName $extension -Publisher $publisher
-Version $ver -PrivateConfiguration $privateConf | Update-AzureVM
```

重置成功后，会将/etc/ssh/sshd\_conf 文件重置为默认配置。

#### 5.4.4 用户自定义脚本扩展

除了上面的扩展外，还可以通过用户自定义脚本扩展来在虚拟机中执行 Powershell 或者 Shell 脚本。用户自定义脚本可以实现非常多的功能，例如通过脚本进行批量部署推送，等等。下面分别演示 Windows 和 Linux 虚拟机通过自定义脚本来实现一些简单的功能。

(1) 使用用户自定义脚本扩展为 Windows 虚拟机添加一个管理员用户。

首先创建一个 adduser.ps1 脚本，脚本内容如下：

```
$computer = "localhost";
$objOu = [ADSI]"WinNT://$computer";
$user = "TestUser";
$password = "Passw0rd!";
$objUser = $objOU.Create("User", $user);
$objUser.setpassword($password);
$objUser.SetInfo();
$objGroup = [ADSI]"WinNT://$computer/Administrators,group";
$objUserAdded = [ADSI]"WinNT://$user";
$objGroup.Add($objUserAdded.PSBase.Path);
```

将脚本上传到存储账号的容器中（容器需要设置为公有容器），要保证公网可以访问到这个脚本（因为扩展实际上也是通过公网去下载这个脚本的）。使用下面的 Azure Powershell 命令在虚拟机中执行上面的脚本：

```
$vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoVM
Set-AzureVMCustomScriptExtension -VM $vm -Run 'adduser.ps1' -FileUri
'http://XXX.blob.core.chinacloudapi.cn/scripts/adduser.ps1'
$vm | Update-AzureVM
```

执行成功后，在虚拟机仪表板中查看扩展的安装情况以及命令的执行情况，如图 5.4-6 所示。

扩展			
名称	版本	状态	消息
Microsoft.Compute.CustomScriptExtension	1.8	Success	Finished executing command

图 5.4-6

到虚拟机中查看，发现已经成功添加了新的用户，如图 5.4-7 所示。

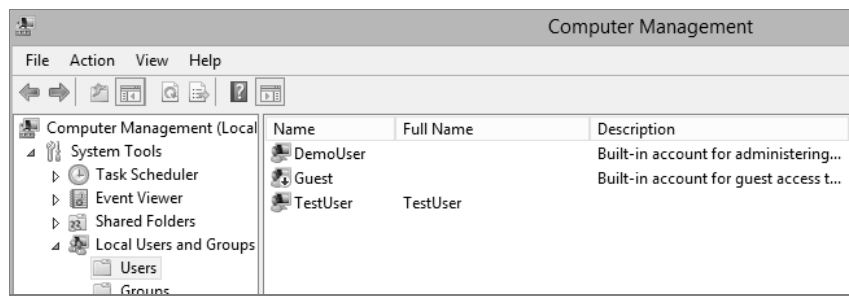


图 5.4-7

(2) 使用用户自定义脚本扩展在 Linux 虚拟机中安装 nginx 服务。  
这里仍然以 CentOS 系统为例，创建一个 install\_nginx.sh 文件，写入下面的内容：

```
#!/bin/sh
touch /etc/yum.repos.d/nginx.repo
echo '[nginx]' >> /etc/yum.repos.d/nginx.repo
echo 'name=nginx repo' >> /etc/yum.repos.d/nginx.repo
echo 'baseurl=http://nginx.org/packages/centos/$releasever/$basearch/' >> /etc/yum.repos.d/nginx.repo
echo 'gpgcheck=0' >> /etc/yum.repos.d/nginx.repo
echo 'enabled=1' >> /etc/yum.repos.d/nginx.repo
yum -y install nginx
```

保存后将文件上传到存储中，同样保证此文件公网可以访问。  
然后使用下面的命令在 Linux 虚拟机中执行上面的安装脚本：

```
$vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoLinuxVM
$PrivateConfiguration = '{ "storageAccountName" : "DemoStorageAccountName" ,
"storageAccountKey" : "<Storage Key>" }'
$PublicConfiguration = '{ "fileUri": [ "http://XXX.blob.core.chinacloudapi.cn/scripts/install_nginx.sh" ], "commandToExecute" : "sh install_nginx.sh" }'
$ExtensionName = 'CustomScriptForLinux'
$Publisher = 'Microsoft.OSTCEExtensions'
$Version = '1.*'
Set-AzureVMExtension -ExtensionName $ExtensionName -VM $vm -Publisher $Publisher -Version $Version -PrivateConfiguration $PrivateConfiguration -PublicConfiguration $PublicConfiguration | Update-AzureVM
```

执行完成后，查看扩展的安装和执行状态，如图 5.4-8 所示。

扩展			
名称	版本	状态	消息
Microsoft.OSTCEExtensions.CustomScriptForLinux	1.5.2.0	Success	Plugin enabled

图 5.4-8

登录虚拟机查看 nginx 的状态，发现已经完成安装。

```
[root@XXX ~]# service nginx status
nginx is stopped
```

除了上面的例子，用户自定义脚本扩展可实现的功能非常多，对于 Windows 虚拟机而言，只要 Powershell 能够实现的功能，理论上都能够使用自定义脚本扩展调用 Powershell 脚本完成。同理，对于 Linux 虚拟机而言，只要 Shell 能够实现的功能，理论上也都能通过该扩展调用 Shell 脚本实现。

### 5.4.5 安全扩展

在 Azure 上可以通过 VM Agent 安装配置 Deep Security 服务，Deep Security 是一款集反恶意软件保护、防火墙、入侵防御系统和完整性监视于一体的软件，关于 Deep Security 更详细的介绍，可以参考链接：<http://blog.trendmicro.com/microsoft-azure-vm-agent-extension-for-deep-security/>。

使用下面的 Azure Powershell 命令为 Windows 虚拟机安装 Deep Security 服务：

```
$vm = Get-AzureVM -ServiceName DemoCloudService -Name DemoVM
$extension = Get-AzureVMExtension -Name AsiaInfo.DeepSecurity
-ExtensionName AsiaInfoDSA
Set-AzureVMExtension -Publisher $extension.Publisher -Version $extension.
Version -ExtensionName $extension.ExtensionName -VM $vm | Update-AzureVM
```

安装完成后，在虚拟机能够找到已安装的 Deep Security 服务，如图 5.4-9 所示。

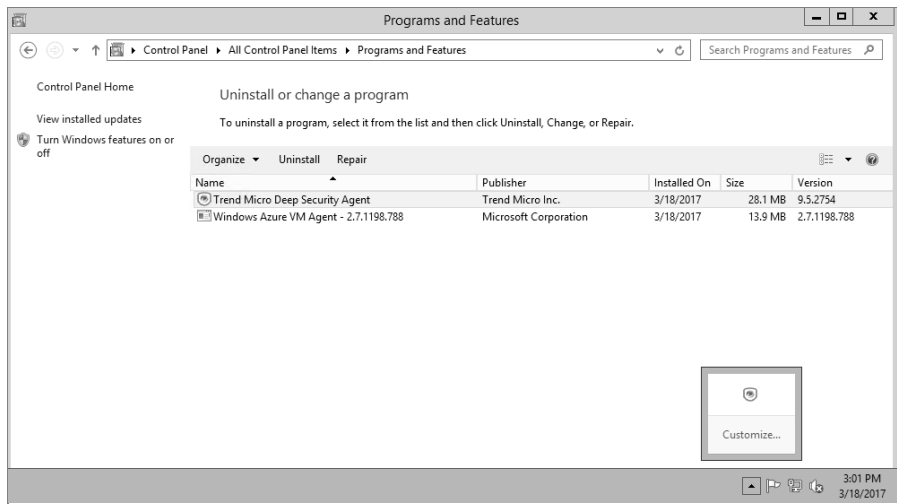


图 5.4-9

## 5.5 多网卡虚拟机的配置和使用

Azure 支持创建多网卡的虚拟机，多网卡的虚拟机是许多网络虚拟设备所必需的。借助多个网卡和网络安全组规则，可以更好地管理网络流量，对多个网卡之间进行流量隔离。

图 5.5-1 显示了具有三个网卡的虚拟机，每个网卡连接到不同的子网，其中默认网卡对应一个 VIP。

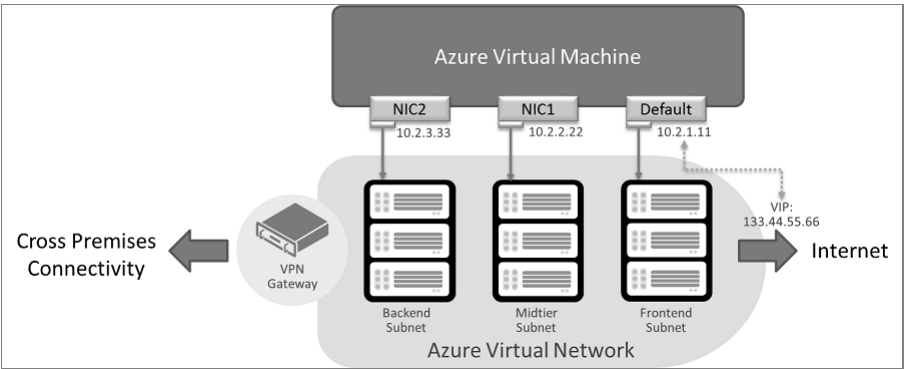


图 5.5-1

不是所有型号的虚拟机都支持多个网卡。虚拟机支持的网卡数，由虚拟机的大小决定。可以参考表 5.5-1 进行选择。

表 5.5-1

最大网卡数	虚 拟 机
2	Standard_A3、Standard_A6、 Standard_A2_v2、Standard_A2m_v2、 Standard_D2、Standard_D11、 Standard_D2_v2、Standard_D11_v2、 Standard_DS2、Standard_DS11、 Standard_DS2_v2、Standard_DS11_v2、 Standard_F2、 Standard_F2s
4	Standard_A4、Standard_A7、 Standard_A4_v2、Standard_A4m_v2、 Standard_D3、Standard_D12、 Standard_D3_v2、Standard_D12_v2、 Standard_DS3、Standard_DS12、 Standard_DS3_v2、Standard_DS12_v2、 Standard_F4、 Standard_F4s
8	Standard_A8_v2、Standard_A8m_v2、 Standard_D4、Standard_D13、Standard_D14、 Standard_D4_v2、Standard_D5_v2、Standard_D13_v2、Standard_D14_v2、Standard_D15_v2、 Standard_DS4、Standard_DS13、Standard_DS14、 Standard_DS4_v2、Standard_DS5_v2、Standard_DS13_v2、Standard_DS14_v2、Standard_DS15_v2、 Standard_F8、Standard_F16、 Standard_F8s、Standard_F16s

创建好的多网卡虚拟机，可以在门户中更改型号配置，但新的虚拟机型号必须能够支



持当前虚拟机实际拥有的网卡数。

在创建虚拟机之前，有以下**限制**要注意：

- 多网卡的虚拟机，必须创建在 Azure 的虚拟网络中；
- 如果不删除和重新创建，则单网卡虚拟机在部署后无法再添加多个网卡（反之亦然）；
- 仅在默认网卡上支持面向 Internet 的公网 IP（VIP）；
- 多网卡虚拟机目前不支持实例级公共 IP（ILPIP）。

如果需要把多网卡的虚拟机放在可用性集中，那么可用性集中的所有虚拟机，都至少需要具备两个网卡。相同的规则也适用于云服务中的虚拟机，因此你可能需要为你的多网卡虚拟机新建一个云服务。但此规则并不要求所有的虚拟机都具有相同数量的网卡。

了解了以上这些限制，就可以开始创建具有多个网卡的虚拟机了。

创建多网卡虚拟机不能在经典管理门户上操作。下面介绍在经典模式下，如何使用 Powershell 创建一台多网卡虚拟机到现有的虚拟网络中。

此处以创建一台 Windows Server 2012 的虚拟机为例，该虚拟机信息如下：

虚拟机型号：超大型（即 Standard\_A4）；

虚拟机系统：Windows Server 2012 Datacenter 中文版；

默认网卡：内网静态 IP “10.0.1.200”，在虚拟网络“multinics”的子网“Sub1”中；

辅助网卡：内网静态 IP “10.0.2.200”，在虚拟网络“multinics”的子网“Sub2”中。  
步骤如下。

登录账户：

```
Add-AzureAccount -Environment AzureChinaCloud
```

获取并查看订阅：

```
Get-AzureSubscription
```

设置默认订阅，-SubscriptionName 是你的订阅名称：

```
Set-AzureSubscription -SubscriptionName "xxxxxxx" -Current
```

也可以使用订阅 ID：

```
Set-AzureSubscription -SubscriptionId xxxxxxxx-Current
```

设置订阅的当前存储账户，-CurrentStorageAccountName 是之前已经创建好的存储账户名称：

```
Set-AzureSubscription -SubscriptionName xxxxxxxx -CurrentStorageAccountName  
xxxxxxx
```

获取名字中包含“Windows-Server-2012”的镜像，并选择只列出镜像名称，用户可以使用 contains 来筛选需要的镜像：

```
Get-AzureVMImage | where {$_.ImageName.Contains(" Windows-Server-2012 ")}  
| select ImageName
```

输出如下：

```

ImageName
-----
0c5c79005aae478e8883bf950a861ce0__Windows-Server-2012-Essentials-20131018
-enus
...
0c5c79005aae478e8883bf950a861ce0__Windows-Server-2012-Essentials-20141204
-zhcn
55bc2b193643443bb879a78bda516fc8__Windows-Server-2012-Datacenter-20151214
-en.us-127GB.vhd
...
55bc2b193643443bb879a78bda516fc8__Windows-Server-2012-Datacenter-20170406
-zh.cn-127GB.vhd
55bc2b193643443bb879a78bda516fc8__Windows-Server-2012-R2-20151214-en.us-
127GB.vhd
...
55bc2b193643443bb879a78bda516fc8__Windows-Server-2012-R2-20170406-zh.cn-
127GB.vhd

```

接着配置虚拟机参数：

指定镜像，以刚才获取到的“55bc2b193643443bb879a78bda516fc8\_Windows-Server-2012-Datacenter-20170406-zh.cn-127GB.vhd”为例：

```
$image = Get-AzureVMImage -ImageName 55bc2b193643443bb879a78bda516fc8_
_Windows-Server-2012-Datacenter-20170406-zh.cn-127GB.vhd
```

设置虚拟机名称及尺寸，-Name 是虚拟机的名称，-InstanceSize 指定虚拟机的型号：

```
$vm = New-AzureVMConfig -Name "multinics" -InstanceSize "ExtraLarge"
-ImageName $image.ImageName
```

设定管理员名称与密码，不要忘记定义系统的类型，-Windows 或-Linux：

```
Add-AzureProvisioningConfig -VM $vm -Windows -AdminUsername "multiuser"
-Password "multiNics2012"
```

添加辅助网卡并指定静态内网 IP，-Name 是网卡名称，-SubnetName 是子网名称，-StaticVNetIPAddress 用来指定静态内网 IP（此处只添加一块网卡，如需添加多块，则修改参数执行此命令多次即可）。每个网卡的地址必须位于一个子网中，但可以向单台虚拟机的多个网卡分配同一个子网中的地址：

```
Add-AzureNetworkInterfaceConfig -Name "NIC2" -SubnetName "Sub2"
-StaticVNetIPAddress "10.0.2.200" -VM $vm
#Add-AzureNetworkInterfaceConfig -Name "NICx" -SubnetName "SubnetName"
-StaticVNetIPAddress "xxx.xxx.xxx.xxx" -VM $vm
```

设置默认网卡子网：

```
Set-AzureSubnet -SubnetNames "Sub1" -VM $vm
```

为默认网卡设置静态 IP：

```
Set-AzureStaticVNetIP -IPAddress "10.0.1.200" -VM $vm
```

以上命令均会输出：

```
AvailabilitySetName          :
ConfigurationSets            : {multinics, Microsoft.WindowsAzure.
Commands.ServiceManagement.Model.NetworkConfigurationSet}
DataVirtualHardDisks        : {}
Label                        : multinics
OSVirtualHardDisk           : Microsoft.WindowsAzure.Commands.
ServiceManagement.Model.OSVirtualHardDisk
RoleName                    : multinics
RoleSize                    : ExtraLarge
RoleType                    : PersistentVMRole
WinRMCertificate            :
X509Certificates            : {}
NoExportPrivateKey          : False
NoRDPEndpoint               : False
NoSSHEndpoint               : False
DefaultWinRmCertificateThumbprint :
ProvisionGuestAgent         : True
ResourceExtensionReferences : {BGInfo}
DataVirtualHardDisksToBeDeleted :
VMImageInput                :
DebugSettings               : Microsoft.WindowsAzure.Commands.
ServiceManagement.Model.DebugSettings
MigrationState              :
LicenseType                 :
```

创建虚拟机，并放在指定的虚拟网络中：

```
New-AzureVM -ServiceName "multinics" -VNetName "multinics" -VMs $vm
```

输出如下：

```
WARNING: No deployment found in service: 'multinics'.
```

OperationDescription	OperationId	OperationStatus
-----	-----	-----
New-AzureVM	5fe2336c-312b-42b9-94a6-94581f8e64b5	Succeeded

创建成功后在门户上可以看到默认网卡信息，如图 5.5-2 所示。



图 5.5-2

远程连接后，查看网卡，可看到多块网卡的信息：

```
Windows IP 配置

以太网适配器 以太网 2:

    连接特定的 DNS 后缀 . . . . . : multinics.a2.internal.chinacloudapp.cn
    本地链接 IPv6 地址. . . . . : fe80::4182:d20a:129f:2637%13
    IPv4 地址 . . . . . : 10.0.1.200
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 10.0.1.1

以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . : multinics.a2.internal.chinacloudapp.cn
    本地链接 IPv6 地址. . . . . : fe80::9431:b16:e187:6de0%12
    IPv4 地址 . . . . . : 10.0.2.200
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :

隧道适配器 isatap.multinics.a2.internal.chinacloudapp.cn:

    媒体状态 . . . . . : 媒体已断开
    连接特定的 DNS 后缀 . . . . . : multinics.a2.internal.chinacloudapp.cn
```

这里要注意，虚拟机内部的网卡顺序是随机的，在 Azure 基础结构更新过程中可能会

更改，但 IP 地址和对应的 MAC 地址不会改变。例如程序中第一段，以太网 2 的 IP 地址为 10.0.1.200，在 Azure 基础结构更新并重启后，以太网 2 的 IP 地址可能会更改为 10.0.2.200，但 IP 和 MAC 的配对则会保持不变。而用户执行的重启，将不会产生这种改变。

默认情况下，辅助网卡上的通信流被限制在同一子网内，并不会配置网关。如果用户希望辅助网卡能与其所在子网之外对话，需要自行添加路由。

## 5.6 云服务的配置和使用

### 5.6.1 云服务 DNS 解析

Azure VM 都位于一个具体的云服务中，可以将云服务理解为虚拟机前端的一台设备，虚拟机的 VIP 实际上是云服务的地址，通过将 VIP 转换为虚拟机内网 IP（做 NAT）和端口映射的方式来实现对虚拟机的访问。对每一个云服务，平台都要求用户定义一个结尾为 .chinacloudapp.cn 的域名，如 test.chinacloudapp.cn。可以在 Azure 经典管理门户的虚拟机仪表板界面查看这个默认的 DNS，如图 5.6-1 所示。



图 5.6-1

尝试解析图 5.6-1 中的默认域名，可以得到 VIP 地址，如图 5.6-2 所示。

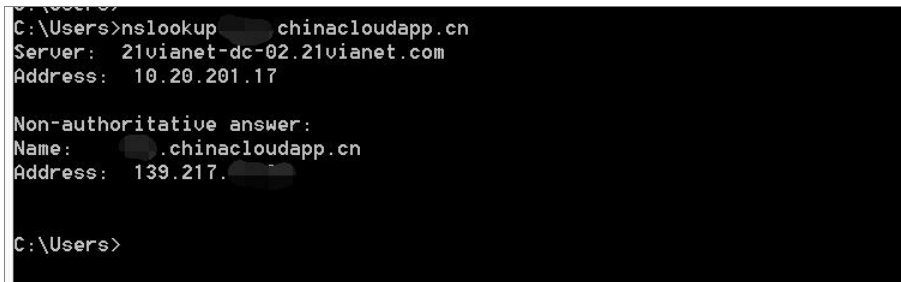


图 5.6-2

在云服务的 VIP 没有固定的情况下，当虚拟机关机再开机后，云服务的 VIP 是有可能发生变化的，但是云服务的 DNS 名称是不会改变的，而且这个 DNS 名称与 VIP 的解析关系由 Azure 平台进行维护，可以始终使用这个 DNS 名称来访问虚拟机。也就是说，当我们

解析这个 DNS 名称时，解析到的地址始终是当前云服务使用的 VIP 而不论这个 VIP 是否已经发生变化。另外如果使用自定义域名来访问虚拟机，可以在 DNS 服务商处做 Cname 映射到云服务的默认域名而不用做 A 记录映射到 VIP。

另外云服务的 VIP 也是可以固定的，如果 VIP 固定了，那么云服务的默认域名每次都是解析到这个固定的 VIP。如果没有必须使用 VIP 访问的需求，正如上文所说，可以直接使用默认域名来访问。

### 5.6.2 实例级公共 IP 地址的 DNS 绑定

Azure 支持给虚拟机单独配置一个实例级公共 IP（Instance PublicIP）地址来实现对虚拟机的访问。配置了实例级公共 IP 地址后，当访问虚拟机时，Azure 仅会将 PublicIP 转换为虚拟机的内网 IP 而不像利用 VIP 访问时还需要做端口的转换。另外配置了 PublicIP 后，对于虚拟机的主动对外请求的流量，源地址默认会使用 PublicIP 而不使用 VIP。在某些环境下，配置 PublicIP 是 Azure 上一种比较好的解决方案，比如虚拟机大量的端口需要打开，向外部的访问非常多，等等，如图 5.6-3 所示。

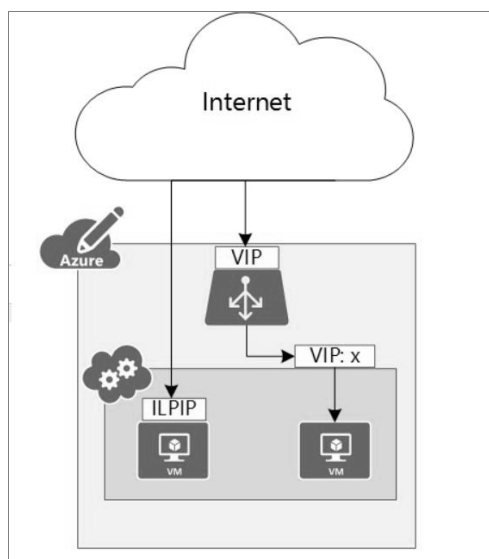


图 5.6-3

给虚拟机配置 PublicIP 不能在经典管理门户上操作，需要使用 Azure Powershell 命令行的方式实现。下面简述操作步骤。

- (1) 选择虚拟机所在的订阅为当前操作订阅。

```
Select-AzureSubscription -SubscriptionName <SubName> -Current
```

- (2) 给虚拟机设置PublicIP，参考图5.6-4。

```
Get-AzureVM -ServiceName <ServiceName> -Name <VMName> | Set-AzurePublicIP  
-PublicIPName <IPName> -DomainNameLabel <DomainPrefix> | Update-AzureVM
```

```
PS C:\Users> Select-AzureSubscription -SubscriptionName " " -Current
PS C:\Users> Get-AzureVM -ServiceName " " -Name " " ; Set-AzurePublicIP -PublicIPName TestPublicIP -DomainNameLabel TestPubIp | Update-AzureVM

OperationDescription      OperationId                OperationStatus
-----
Update-AzureVM            7812e0d7-447f-4695-93b0-87e517fc7037 Succeeded

PS C:\Users>
```

图 5.6-4

其中-DomainNameLabel 是指明需要设置的 DNS 域名的前缀，添加了这个参数，可以使用 Get-AzureVM 命令查看到这个 PIP 地址以及关联了<DomainPrefix>.<云服务名称>.chinacloudapp.cn 的域名，如图 5.6-5 所示。

查看虚拟机的 PublicIP:

```
Get-AzureVM -ServiceName <ServiceName> -Name <VMName>
```

```
PS C:\Users> Get-AzureVM -ServiceName " " -Name " "

DeploymentName      : 
Name               : 
Label              : 
VM                 : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.PersistentVM
InstanceStatus      : ReadyRole
IpAddress           : 10.20.0.4
InstanceStateDetails : 
PowerState          : Started
InstanceErrorCode    : 
InstanceFaultDomain : 0
InstanceName        : 
InstanceUpgradeDomain : 0
InstanceSize        : Large
HostName            : 
AvailabilitySetName  : 
DNSName             : 
Status              : ReadyRole
GuestAgentStatus     : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.GuestAgentStatus
ResourceExtensionStatusList : {Microsoft.Compute.BGInfo}
PublicIPAddress      : 139.217.
PublicIPName        : TestPublicIP
PublicIPDomainNameLabel : TestPubIp
PublicIPFqdns        : {TestPubIp.fuz1bjb05.chinacloudapp.cn, TestPubIp.0.fuz1bjb05.chinacloudapp.cn}
NetworkInterfaces    : {}
VirtualNetworkName   : 
RemoteAccessCertificateThumbprint : 693CA6BA486742D5F56EF14215F1350AE168D311
ServiceName          : 
OperationDescription  : Get-AzureVM
OperationId           : 4477f9afc6c14f778086297c43c2d59d
OperationStatus       : OK
```

图 5.6-5

从图 5.6-5 输出的结果可以看到，PublicIPFqdns 显示了-DomainNameLabel 参数设置的域名。因为 PublicIP 是无法固定的，一旦虚拟机关机、重启就有可能发生改变，因此最好给这个 IP 设置一个默认域名，因为默认域名是不会改变的，它会始终解析到当前虚拟机使用的 PublicIP，可参见下面的解析测试，如图 5.6-6 所示。

另外如果有自定义域名的话，可以在 DNS 服务商处设置 Cname 解析到这个默认域名来实现对虚拟机的访问，这样可以避免 PublicIP 不能固定的问题。

```
C:\Users>nslookup TestPubIp.fuz1bjb05.chinacloudapp.cn
Server: 21vianet-dc-02.21vianet.com
Address: 10.20.201.17

Non-authoritative answer:
Name: TestPubIp.fuz1bjb05.chinacloudapp.cn
Address: 139.217.121.159

C:\Users>
```

图 5.6-6

### 5.6.3 反向 DNS 解析

Azure 支持对云服务的域名做反向 DNS 解析，也就是说可以利用 IP 反向查找其对应的域名。另外对于 Cname 到云服务域名的自定义域名，也可以设置反向域名解析，反向域名解析的应用场景一般在邮件服务中。下面介绍如何设置域名的反向解析。

#### 1. 设置云服务域名的反向解析

设置云服务域名的反向解析需要使用 Azure Powershell 命令，如图 5.6-7 所示。

```
Set-AzureService -ServiceName <ServiceName> -Description <SelfDescription>
-ReverseDnsFqdn <xxx.chinacloudapp.cn.>
```

请注意域名的结尾一定要加上.作为后缀。

```
PS C:\Users\<User>> Set-AzureService -ServiceName <ServiceName> -Description "ReverseDNS" -ReverseDnsFqdn "<ServiceName>.chinacloudapp.cn."
```

OperationDescription	OperationId	OperationStatus
Set-AzureService	f3ace270-9e7e-4575-8a08-ccd08046f642	Succeeded

```
PS C:\Users\<User>>
```

图 5.6-7

验证上面的设置是否生效，可以使用下面命令，如图 5.6-8 所示。

```
nslookup -type=ptr VIP
```

```
C:\Users>nslookup -type=ptr 42.159.121.159
Server: 21vianet-dc-02.21vianet.com
Address: 10.20.201.17

Non-authoritative answer:
42.159.121.in-addr.arpa name = <ServiceName>.chinacloudapp.cn

C:\Users>
```

图 5.6-8



## 2. 设置自定义域名的反向解析

假如有一个自定义域名 `test.2dream.com.cn`，在 DNS 服务商那里做了 `cname` 记录映射到云服务域名 `xxx.chinacloudapp.cn`，如图 5.6-9 所示。



图 5.6-9

接下来就可以设置这个自定义域名的反向解析，如图 5.6-10 所示。

```
Set-AzureService -ServiceName <ServiceName> -Description <SelfDescription>
-ReverseDnsFqdn "test.2dream.com.cn."
```



图 5.6-10

测试上面设置的自定义域名的反向解析，如图 5.6-11 所示。



图 5.6-11

另外需要注意的是，在做自定义域名的反向解析的时候，必须满足下面的两个条件之一：

(1) 自定义域名必须能解析到（也就是 CName 到）同订阅下的任何一个云服务的 DNS 名称 `xxxx.chinacloudapp.cn`。

(2) 自定义域名必须能解析到(也就是 A 记录)同订阅下的任何一个云服务的 VIP，否则会报出错误，说明自定义域名必须能解析到云服务的域名或者 VIP，如图 5.6-12 所示。

```
PS C:\Users> Set-AzureService -ServiceName SelfReverseDNS -Description "SelfReverseDNS" -ReverseDnsFqdn "test.2dream.com.cn."
Set-AzureService : BadRequest: The reverse DNS FQDN test.2dream.com.cn. must resolve to one of: a). the DNS name of this Hosted Service (fuzibjb03.chinacloudapp.cn), b). the DNS name of a different Hosted Service in this subscription (eb385e00-d509-40dd-8372-6cb3a2ff51eb), c). a Reserved IP belonging to this subscription, or d). the IP of a deployment or of a VM in this subscription.
OperationID : '2f37995097664761bac022044d7d5895'
At line:1 char:1
Set-AzureService -ServiceName fuzibjb03 -Description "SelfReverseDNS" -ReverseDnsFqdn "test.2dream.com.cn."
+ CategoryInfo          : (CloseError: ({})) [Set-AzureService]. ComputeCloudException
+ FullyQualifiedErrorId : Microsoft.WindowsAzure.Commands.ServiceManagement.HostedServices.SetAzureServiceCommand
```

图 5.6-12

#### 5.6.4 云服务多 VIP 配置参考

云服务的 VIP 实际上是关联到虚拟机前端的 Azure Load Balancer，而不是关联到云服务中的虚拟机实例。通过将 VIP 转换为虚拟机的内网 IP 和将公网端口转换为虚拟机内部端口，我们可以使用单个 VIP 访问云服务中的任何 VM 实例。

但是，在某些情况下，你可能需要多个 VIP 作为同一云服务的入口点。例如，云服务中有多个虚拟机，每个虚拟机都托管了一个网站，而每一个网站都需要使用默认端口 80 来访问，因为每个站点是针对不同的客户或租户托管的。在此情况下，每个网站都需要有不同的面向公众的 IP 地址。图 5.6-13 阐明了一个典型的在云服务上关联多个 VIP 的环境。

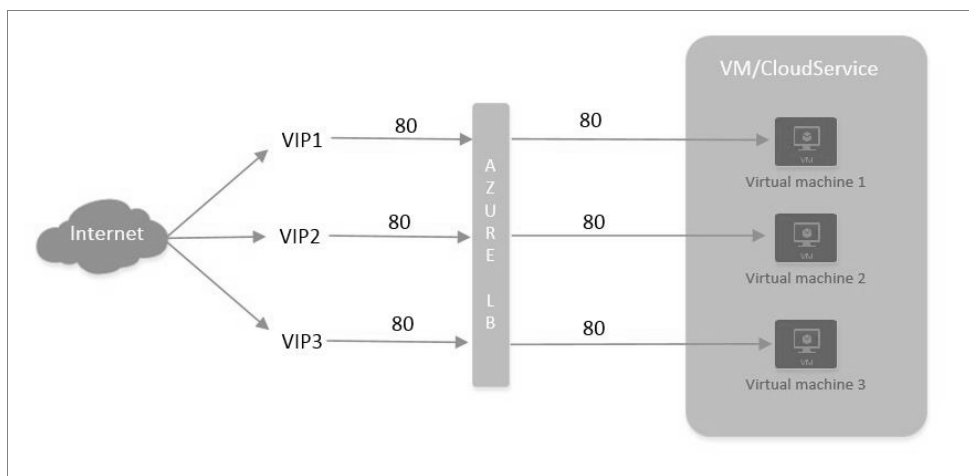


图 5.6-13

在上面的示例中，所有 VIP 使用相同的公共端口（80），将访问流量定向到后端托管网站的每一个 VM 上。下面讲述如何将 VIP 添加到云服务，需要使用 Azure Powershell 来操作，不支持在经典管理门户上进行操作。

首先使用如下 Powershell 命令：

```
$dep=Get-AzureDeployment -ServiceName <ServiceName>
$dep.VirtualIPs
```

查看一下云服务的默认 VIP，图 5.6-14 中命令示例的 VIP 是 139.217.26.194。

```
PS C:\Users> $dep=Get-AzureDeployment -ServiceName <ServiceName>
PS C:\Users> $dep.VirtualIPs

Address      : 139.217.26.194
IsDnsProgrammed : True
Name         : fuz1bjb05winContractContract
ReservedIPName :
ExtensionData :
```

图 5.6-14

使用 Powershell 命令给云服务添加一个 VIP。

```
Add-AzureVirtualIP -ServiceName <ServiceName> -VirtualIPName <VipName>
```

其中-VirtualIPName 指明这个添加的 VIP 名称，图 5.6-15 的示例命令将 VIP 命名为 VIP1。

```
PS C:\Users> Add-AzureVirtualIP -ServiceName <ServiceName> -VirtualIPName VIP1

OperationDescription      OperationId      OperationStatus
-----
Add-AzureVirtualIP        44337d16-494b-476e-a60b-6084e1ea986d      Succeeded

PS C:\Users> $dep=Get-AzureDeployment -ServiceName <ServiceName>
PS C:\Users> $dep.VirtualIPs

Address      : 139.217.26.194
IsDnsProgrammed : True
Name         : fuz1bjb05winContractContract
ReservedIPName :
ExtensionData :

Address      :
IsDnsProgrammed :
Name         : VIP1
ReservedIPName :
ExtensionData :
```

图 5.6-15

从图 5.6-15 的输出看，确实可以看到云服务已经添加了 VIP1，但是 VIP1 并没有一个实际的地址，因为该 VIP 并没有关联的具体终结点，只有关联了具体的终结点后，才能具体看到该 VIP 是多少。下面将一个具体的终结点关联到该 VIP，并且该终结点也是关联到该云服务中的一台虚拟机的，使用的命令如下，命令示例可参考图 5.6-16。

```
Get-AzureVM -ServiceName <ServiceName> -Name <VMname> | Add-AzureEndpoint
-Name <EndpointName> -Protocol tcp -LocalPort <LocalPortNumber> -PublicPort
<PublicPortNumber> -VirtualIPName <VipName> | Update-AzureVM
```

```
PS C:\Users> Get-AzureVM -ServiceName [redacted] -Name [redacted] | Add-AzureEndpoint -Name httpone -Protocol tcp -LocalPort 80 -PublicPort 80 -VirtualIPName VIP1 | Update-AzureVM
```

OperationDescription	OperationId	OperationStatus
Update-AzureVM	8907dd83-c483-4191-a7ec-13aebd9818f8	Succeeded

图 5.6-16

说明：-Name 指的是为这个终结点自定义的名称；-Protocol 指定终结点采用的协议，一般为 TCP 或者 UDP；-LocalPort 指的是虚拟机所开的内网端口；-Publicport 指的是公网端口；-VirtualIpname 指的是将这个终结点关联到哪个 VIP，这里当然填写 VIP1 了。

运行上面的命令并再次查看云服务的 VIP，可以看到 VIP1 的地址，参考图 5.6-17。

```
PS C:\Users> $dep=Get-AzureDeployment -ServiceName [redacted]
PS C:\Users> $dep.VirtualIPs
```

Address	: 139.217.26.194
IsDnsProgrammed	: True
Name	: fuzljbjb05winContractContract
ReservedIPName	:
ExtensionData	:
Address	: 139.217.13.154
IsDnsProgrammed	:
Name	: VIP1
ReservedIPName	:
ExtensionData	:

```
PS C:\Users>
```

图 5.6-17

在经典管理门户虚拟机的终结点部分可以看到添加的终结点，参考图 5.6-18。



名称	协议	公用端口	私有端口	负载均衡名称
SSH	TCP	22	22	-
httpone	TCP	80	80	-

图 5.6-18

如需要为云服务再关联新的 VIP，按照以上步骤重复操作即可。需要注意的是，云服务的默认 DNS 名称 xxx.chinacloudapp.cn 只会解析到云服务最初的 VIP，而不会解析到我们手动关联的 VIP。参考下面的解析测试，如图 5.6-19 所示。另外从图 5.6-17 中可以看到，只有默认 VIP 的 IsDnsProgrammed 属性值为 True。

```

C:\Users>
C:\Users>nslookup [REDACTED].chinacloudapp.cn
Server: 21vianet-dc-02.21vianet.com
Address: 10.20.201.17

Non-authoritative answer:
Name: [REDACTED].chinacloudapp.cn
Address: 139.217.26.194

C:\Users>

```

图 5.6-19

与在云服务默认的 VIP 上启用负载均衡设置一样，也可以在手动添加的 VIP 上启动负载均衡。例如，针对上面添加的 VIP1，希望将发往 VIP1 的公网端口 8080、内网端口 8080 的流量在两个后端虚拟机之间进行负载均衡，可以使用下面的命令并参考图 5.6-20。

```

Get-AzureVM -ServiceName <ServiceName> -Name <VmName> | Add-AzureEndpoint
-Name myendpoint -LoadBalancedEndpointSetName LBTest -Protocol tcp -LocalPort
8080 -PublicPort 8080 -VirtualIPName VIP1 -DefaultProbe | Update-AzureVM
Get-AzureVM -ServiceName <ServiceName> -Name <VmName> | Add-AzureEndpoint
-Name myendpoint -LoadBalancedEndpointSetName LBTest -Protocol tcp -LocalPort
8080 -PublicPort 8080 -VirtualIPName VIP1 -DefaultProbe | Update-AzureVM

```

```

PS C:\Users> Get-AzureVM -ServiceName [REDACTED] -Name [REDACTED] | Add-AzureEndpoint -Name myendpoint -LoadBalancedEndpointSetName LBTest -Protocol tcp -LocalPort 8080 -PublicPort 8080 -VirtualIPName VIP1 -DefaultProbe | Update-AzureVM

```

OperationDescription	OperationId	OperationStatus
Update-AzureVM	cbca1530-29d1-4334-950a-6f0b17aec502	Succeeded

```

PS C:\Users> Get-AzureVM -ServiceName [REDACTED] -Name [REDACTED] | Add-AzureEndpoint -Name myendpoint -LoadBalancedEndpointSetName LBTest -Protocol tcp -LocalPort 8080 -PublicPort 8080 -VirtualIPName VIP1 -DefaultProbe | Update-AzureVM

```

OperationDescription	OperationId	OperationStatus
Update-AzureVM	99a3a01e-cd8e-4df9-9b87-39410a60601c	Succeeded

```

PS C:\Users>

```

图 5.6-20

添加完成之后，可以在经典管理门户虚拟机的终结点界面看到设置的负载均衡集，参考图 5.6-21。

	仪表板 监视器 终结点 配置				
	名称	协议	公用端口	私有端口	负载均衡集名称
	SSH	TCP	22	22	-
	httpone	TCP	80	80	-
	myendpoint	TCP	8080	8080	LBTest

图 5.6-21

手动添加的 VIP 如果需要保留，防止虚拟机关机、开机后发生变化，可以使用下面的命令进行固定，示例可参考图 5.6-22。

```
New-AzureReservedIP -ServiceName <ServiceName> -ReservedIPName VIPResv
-Location "China North" -VirtualIPName <VipName>
```

说明：-ReservedIPName 指明将这个 VIP 设置为某个保留名称，可自定义；  
-Location 为 China East 或者 China North，依据虚拟机的位置而定；  
-VirtualIPName 就是填写要保留的 VIP 名称，这里当然是 VIP1 了。

```
PS C:\Users> New-AzureReservedIP -ServiceName <ServiceName> -ReservedIPName VIPResv -Location "China North" -VirtualIPName VIP1

OperationDescription      OperationId                OperationStatus
-----
New-AzureReservedIP      23de15d7-6096-4e74-8b3c-8fe9e2296539 Succeeded

PS C:\Users> $dep-Get-AzureDeployment -ServiceName <ServiceName>
PS C:\Users> $dep.VirtualIPs

Address      : 139.217.26.194
IsDnsProgrammed : True
Name         : fuzljb05winContractContract
ReservedIPName :
ExtensionData :
Address      : 139.217.13.154
IsDnsProgrammed :
Name         : VIP1
ReservedIPName : VIPResv
ExtensionData :
```

图 5.6-22

最后需要注意的是下面两点：

(1) 未关联到额外添加的 VIP 的终结点是使用默认的 VIP 来访问。使用默认的终结点无法访问关联到特定 VIP 的终结点。比如无法使用默认的 VIP 来访问关联到 VIP1 的 80 端口，如图 5.6-23 所示。

```
C:\Users> psping.exe 139.217.26.194:80

PsPing v2.01 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2014 Mark Russinovich
Sysinternals - www.sysinternals.com

TCP connect to 139.217.26.194:80:
5 iterations (warmup 1) connecting test:
Connecting to 139.217.26.194:80 (warmup): This operation returned because the ti
meout period expired.
Connecting to 139.217.26.194:80: This operation returned because the timeout per
iod expired.
Connecting to 139.217.26.194:80: This operation returned because the timeout per
iod expired.
Connecting to 139.217.26.194:80: This operation returned because the timeout per
iod expired.
Connecting to 139.217.26.194:80: This operation returned because the timeout per
iod expired.
```

图 5.6-23

(2) 配置了额外的 VIP 后，无法通过界面去调整虚拟机的配置了，例如在界面中调整虚拟机大小或者可用性集，或者添加、删除终结点，等等，会提示错误，参考图 5.6-24。只能使用 Powershell 去操作这些设置。

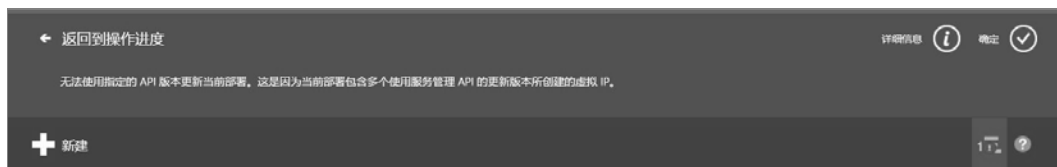


图 5.6-24

# 第六章 存 储

本章主要介绍了普通存储和高级存储的使用方法、性能优化、一些常见问题的总结和  
处理方法、文件共享服务的使用方法、存储管理工具的安装和使用，以及磁盘阵列和 LVM  
的配置方法。

## 6.1 文件存储常见问题

### 1. 文件存储是否支持基于 Active Directory 的身份验证？

我们目前不支持基于 AD 的身份验证或 ACL，但会将其列入我们的功能请求列表中。  
目前，Azure 存储账户密钥用于为文件共享提供身份验证。我们提供通过 REST API 或客户  
端库使用共享访问签名（SAS）的解决方法。使用 SAS，可以生成具有在指定的时间间隔  
内有效的特定权限的令牌。例如，你可以生成对给定文件具有只读访问权限的令牌。在此  
令牌的有效期内拥有此令牌的任何人对该文件具有只读访问权限。

仅通过 REST API 或客户端库支持 SAS。通过 SMB 协议装载文件共享时，不能使用  
SAS 委派对其内容的访问权限。

### 2. 如何通过 Web 浏览器提供对特定文件的访问权限？

使用 SAS，可以生成具有在指定的时间间隔内有效的特定权限的令牌。例如，可以生  
成一个令牌，在特定的时段对特定的文件进行只读访问。只要该 URL 有效，任何人都可以  
使用它直接从任何 Web 浏览器进行下载。可以轻松地通过 UI（例如存储资源管理器）生  
成 SAS 密钥。

### 3. 访问 Azure 文件存储中的文件可以通过哪些不同方式？

可以使用 SMB 3.0 协议将文件共享装载在本地计算机上，也可以使用存储资源管理器  
或 Cloudberry 之类的工具访问文件共享中的文件。可以通过应用程序使用客户端库、REST  
API 或 Powershell 访问 Azure 文件共享中的文件。

### 4. 如何在本地计算机上装载 Azure 文件共享？

可以通过 SMB 协议装载文件共享，只要端口 445（TCP 出站）处于打开状态且客户端  
支持 SMB 3.0 协议（例如，Windows 8 或 Windows Server 2012）。请通过本地 ISP 提供商  
来取消阻止端口。在过渡期间，可以使用存储资源管理器或任何其他第三方（例如  
Cloudberry）查看文件。由于 Linux SMB 尚不支持加密，从 Linux 文件共享仍需要客户端  
与文件共享在同一 Azure 中。但是，Linux 加密支持已经在负责 SMB 的 Linux 人员的路线  
图上。将来支持加密的 Linux 也将能够从任何位置装载 Azure 共享。



**5. Azure 虚拟机与文件共享之间的网络流量是否算作对订阅计费的外部带宽？**

如果文件共享和虚拟机位于不同的区域，则它们之间的流量将作为外部带宽收费。

**6. 如果是虚拟机和同一区域中的文件共享之间的网络流量，是免费吗？**

是的。如果流量在同一区域，是免费的。

**7. 从本地虚拟机连接到 Azure 文件存储是否依赖于 Azure ExpressRoute？**

不能。如果没有 ExpressRoute，仍可从本地访问文件共享，只需将端口 445（TCP 出站）打开供 Internet 访问即可。但是，如果你愿意，你可以将 ExpressRoute 用于文件存储。

**8. 故障转移群集的“文件共享见证”是 Azure 文件存储的使用案例之一吗？**

目前不支持此功能。

**9. 当前仅通过 LRS 或 GRS 复制文件存储，对吗？**

我们计划支持 RA-GRS，但具体时间尚未确定。

**10. 何时能够将现有存储账户用于 Azure 文件存储？**

现已为所有存储账户启用 Azure 文件存储。

**11. 是否会将重命名操作也添加到 REST API？**

在我们的 REST API 中尚不支持重命名。

**12. 能否使用嵌套共享，换言之就是共享下的共享？**

否。文件共享是你装载的虚拟驱动程序，因此不支持嵌套共享。

**13. 是否可以对共享中的文件夹指定只读或只写权限？**

如果通过 SMB 装载文件共享，你不具有此级别的权限控制。但是，你可以通过 REST API 或客户端库创建共享访问签名（SAS）来实现此控制。

**14. 尝试将文件解压缩到文件存储中时我的性能速度太慢。我该怎样做？**

若要将大量文件传输到文件存储，建议使用 AzCopy、Azure Powershell（Windows）或 Azure CLI（Linux/Unix），因为这些工具已针对网络传输进行优化。

**15. 发布了修复 Azure 文件慢速性能问题的修补程序**

Windows 团队最近发布了一个修补程序，旨在修复客户从 Windows 8.1 计算机或 Windows Server 2012 R2 服务器访问 Azure 文件存储时遇到的慢速性能问题。有关详细信息，请查看相关的知识库文章：从 Windows 8.1 或 Server 2012 R2 访问 Azure 文件存储时性能降低。

**16. 通过 IBM MQ 使用 Azure 文件存储**

IBM 已发布相关文档来指导 IBM MQ 客户通过其服务配置 Azure 文件存储。有关详细信息，请查阅如何通过 Azure 文件服务设置 IBM MQ 多实例队列管理器。

### 17. 如何排除 Azure 文件存储错误？

可以参考 Azure 文件故障排除文章了解有关端到端故障排除指南。

### 18. 如何针对 Azure 文件启用服务器端加密？

针对 Azure 文件的服务器端加密目前提供预览版。在预览期间，只能在使用 Azure 门户预览新建的 Azure Resource Manager 存储账户上启用此功能。启用该功能没有额外收费。针对 Azure 文件存储启用存储服务加密以后，系统会自动加密数据。

我们计划未来允许用户通过 Azure PowerShell、Azure CLI 和 Azure 存储资源提供程序 REST API 为文件存储启用加密。若要详细了解如何在 Azure 存储中进行静态加密，请参阅存储服务加密。

## 6.2 Azure 文件存储问题疑难解答

本文列出了从 Windows 和 Linux 客户端连接时与 Azure 文件存储相关的常见问题。它还提供了这些问题的可能原因和解决方法。

### 常规问题（在 Windows 和 Linux 客户端中均存在）

- 尝试打开文件时配额出错。
- 从 Windows 或 Linux 访问 Azure 文件存储时性能不佳。
- 如何跟踪 Azure 文件存储中的读写操作。

### Windows 客户端问题

- 从 Windows 8.1 或 Windows Server 2012 R2 访问 Azure 文件存储时性能不佳。
- 尝试装载 Azure 文件共享时出现错误 53。
- 尝试装载 Azure 文件共享时出现错误 87：参数不正确。
- Net use 成功，但未显示装载在 Windows 资源管理器上的 Azure 文件共享。
- 我的存储账户包含 “/” 且 net use 命令失败。
- 我的应用程序/服务无法访问装载的 Azure 文件驱动器。
- 其他性能优化建议

### Linux 客户端问题

- 将文件上传/复制到 Azure 文件时出现错误“正在将文件复制到不支持加密的目标”。
- 间歇性 IO 错误 - 在装载点上执行列表命令时，现有的文件共享上出现错误“主机已关闭”或外壳挂起。
- 尝试在 Linux VM 上装载 Azure 文件时出现装入错误 115。
- Linux VM 在类似 “ls” 的命令中遇到随机延迟。
- 错误 112 - 超时错误。

### 从其他应用程序访问

- 是否可以通过 Web 作业引用应用程序的 Azure 文件共享？

### 6.2.1 尝试打开文件时配额出错

在 Windows 中，将收到类似下文的错误消息：

```
1816 ERROR_NOT_ENOUGH_QUOTA <--> 0xc0000044 STATUS_QUOTA_EXCEEDEDNot enough quota is available to process this command Invalid handle value GetLastError: 53
```

在 Linux 中，将收到类似下文的错误消息：

```
<filename> [permission denied] Disk quota exceeded
```

#### 原因

问题原因是已达到文件所允许的并发打开句柄数上限。

#### 解决方案

关闭某些句柄以减少并发打开句柄数，然后重试。有关详细信息，请参阅 [Azure 存储性能和可伸缩性核对清单](#)。

### 6.2.2 从 Windows 或 Linux 访问文件存储时性能不佳

如果没有特定的 I/O 大小下限要求，建议使用 1 MB 的 I/O 大小获得最佳性能。

如果知道使用写入扩展的文件的最终大小，并且当尚未在文件上写入的尾部包含零时软件没有兼容性问题，请提前设置文件大小，而不是每次写入都是扩展写入。

使用正确的复制方法：

- 使用 AZCopy 在两个文件共享之间进行任何传输活动。有关更多详细信息，请参阅 [使用 AzCopy 命令行实用工具传输数据](#)。
- 在文件共享与本地计算机之间使用 Robocopy。有关详细信息，请参阅 [多线程 Robocopy 加快复制速度](#)。

### 6.2.3 从 Windows 8.1 或 Windows Server 2012 R2 访问文件存储时的性能不佳

对于运行 Windows 8.1 或 Windows Server 2012 R2 的客户端，请确保安装有修补程序 KB3114025。该程序可提升创建和关闭句柄时的性能。

可运行以下脚本，检查是否安装了该修补程序：

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters\Policies
```

如果已安装，将显示以下输出：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters\Policies {96c345ef-3cac-477b-8fcd-bea1a564241c} REG_DWORD 0x1
```

#### 如何跟踪 Azure 文件存储中的读写操作

Microsoft Message Analyzer 能够以明文形式显示客户端的请求，并且有线请求和事务

之间的关系良好（假设此处的 SMB 不是 REST）。其缺点在于，如果存在许多 IaaS VM 工作进程，则需要在每个客户端上运行此工具，因此十分耗时。

如果将 Message Analyze 与 ProcMon 配合使用，就可以清楚了解负责事务的应用代码。

## 6.2.4 其他性能优化建议

请勿为请求写入权限但不允许读取权限的缓存 I/O 创建或打开文件。这就是说，当调用 `CreateFile()` 时，永远不要只指定 `GENERIC_WRITE`，而是要始终指定 `GENERIC_READ | GENERIC_WRITE`。只写句柄无法在本地缓存小型写入，即使它是文件唯一打开的句柄。这会严重影响小型写入操作的性能。请注意，指向 `CRT fopen()` 的“a”模式会打开只写句柄。

## 6.2.5 尝试装载或卸载 Azure 文件共享时出现“错误 53”或“错误 67”

此问题的原因可能如下。

### 原因 1

“发生系统错误 53。访问被拒绝。”出于安全原因，如果信道未加密，且未从 Azure 文件共享所驻留的数据中心尝试连接，则到 Azure 文件共享的连接将受阻。如果用户的客户端 OS 不支持 SMB 加密，则不会加密信道。当用户尝试从本地或其他数据中心装载文件共享时，这将表示为“发生系统错误 53。访问被拒绝”错误消息。Windows 8、Windows Server 2012 及更高版本的每次协商均要求其包含支持加密的 SMB 3.0。

### 原因 1 的解决方案

通过满足 Windows 8、Windows Server 2012 或更高版本要求的客户端进行连接，或用于连接的虚拟机要与 Azure 文件共享所用的 Azure 存储账户位于同一数据中心。

### 原因 2

如果端口 445 到 Azure 文件数据中心的出站通信受阻，则在安装 Azure 文件共享时可能会出现“系统错误 53”或“系统错误 67”。请单击此处，概要查看允许或禁止从端口 445 进行访问的 ISP。

Comcast 和某些 IT 组织阻止此端口。若要了解是否由此造成“系统错误 53”，可使用 Portqry 查询 TCP:445 终结点。如果 TCP:445 终结点显示为“已筛选”，则表示 TCP 端口受阻。示例查询如下：

```
g:\DataDump\Tools\Portqry>PortQry.exe -n [storage account name].file.core.chinacloudapi.cn -p TCP -e 445
```

如果 TCP 445 受到网络路径中的规则阻止，将显示以下输出：

### TCP 端口（445microsoft-ds 服务）：筛选

若要深入了解如何使用 Portqry，请参阅 Portqry.exe 命令行实用工具说明。

### 原因 2 的解决方案

与 IT 组织配合，向 Azure IP 范围开放端口 445 出站。

### 原因 3

如果在客户端上启用 NTLMv1 通信，也会收到“系统错误 53 或系统错误 87”。启用 NTLMv1 会降低客户端的安全性。因此，将阻止 Azure 文件的通信。若要验证这是否是错误原因，请验证以下注册表子项是否设为值 3：

HKLM\SYSTEM\CurrentControlSet\Control\Lsa > LmCompatibilityLevel。

有关详细信息，请参阅 TechNet 上的 LmCompatibilityLevel 主题。

### 原因 3 的解决方案

若要解决此问题，请将 HKLM\SYSTEM\CurrentControlSet\Control\Lsa 注册表项中的 LmCompatibilityLevel 值恢复为默认值 3。

Azure 文件仅支持 NTLMv2 身份验证。请确保客户端上应用了组策略。这将防止发生此错误。这也被视为最佳安全方案。有关详细信息，请参阅如何通过组策略配置客户端以使用 NTLMv2

建议策略设置为**仅发送 NTLMv2 响应**。这对应于注册表值为 3。客户端仅使用 NTLMv2 身份验证；如果服务器支持，则使用 NTLMv2 会话安全。域控制器接受 LM、NTLM 和 NTLMv2 身份验证。

## 6.2.6 Net use 成功，但未显示装载在 Windows 资源管理器上的 Azure 文件共享

### 原因

默认情况下，Windows 资源管理器不以管理员身份运行。如果通过 Administrator 命令提示符运行 **net use**，会将网络驱动器映射为“管理员身份”。由于映射的驱动器以用户为中心，如果在其他用户账户上安装了这些驱动器，则登录的用户账户不会显示它们。

### 解决方案

通过非管理员命令行中装载共享。或者，可按照此 TechNet 主题配置 **EnableLinked Connections** 注册表值。

## 6.2.7 我的存储账户包含“/”且 net use 命令失败

### 原因

当 **net use** 命令在命令提示符 (cmd.exe) 下运行时，添加“/”作为命令行选项对其进行解析。这会导致驱动器映射失败。

### 解决方案

可使用下述某个步骤解决此问题：

- 使用以下 PowerShell 命令：

```
New-SmbMapping -LocalPath y: -RemotePath \\server\share -UserName accountName -Password "password can contain / and \ etc"
可在批处理文件中执行以下命令
Echo new-smbMapping ... | powershell -command -
```

- 将密钥用双引号括起以解决此问题（除非第一个字符是“/”）。若是此例外，则使用交互模式，然后单独输入密码或重新生成密钥，获取不以正斜杠（/）字符开头的密钥。

## 6.2.8 我的应用程序/服务无法访问装载的 Azure 文件驱动器

### 原因

根据用户装载驱动器。如果应用程序或服务在其他用户账户下运行，则驱动器不可见。

### 解决方案

从应用程序所在的同一用户账户下载载驱动器。可使用 `psexec` 工具完成此操作。

或者，可创建具有网络服务或系统账户相同特权的新用户，然后运行该账户下的 `cmdkey` 和 `net use`。用户名称应为存储账户名称，密码应为存储账户密钥。还有一种针对 `net use` 的方法，就是传入 `net use` 命令的用户名和密码参数中的存储账户名及密钥。

按照说明操作后，在为系统/网络服务账户运行 `net use` 时，可能会收到以下错误消息：“发生系统错误 1312。指定的登录会话不存在。其可能已终止”。若发生此情况，请确保传递到 `net use` 的用户名包含域信息（例如 “[存储账户名].file.core.chinacloudapi.cn”）。

## 6.2.9 “正在将文件复制到不支持加密的目标” 错误

### 原因

可将 Bitlocker 加密的文件复制到 Azure 文件。但文件存储不支持 NTFS EFS。因此，此情况下可能要使用 EFS。如果通过 EFS 加密文件，则可能无法复制到文件存储，除非复制命令将解密复制后的文件。

### 解决方法

必须先解密，才能将文件复制到文件存储。为此，可执行下述某种方法：

- 使用 `copy /d`。
- 设置以下注册表项：

- 路径=HKLM\Software\Policies\Microsoft\Windows\System
- 值类型=DWORD
- 名称= CopyFileAllowDecryptedRemoteDestination
- 值= 1

但请注意，设置注册表项会影响所有到网络共享的复制操作。

## 6.2.10 现有文件共享上出现“主机已关闭”错误，或者在装入点上运行列表命令时 shell 挂起

### 原因

客户端长时间处于空闲状态时，Linux 客户端上将出现此错误。此错误发生时，客户端断开连接且客户端连接超时。

### 解决方案

Linux 内核的更改集中修复了此问题，该更改集正在等待移植到 Linux 分发中。

为了解决此问题，请在 Azure 文件共享中保留一个定期写入的文件，保持连接并避免进入空闲状态。此操作必须为写入操作，例如在文件上重新写入创建/修改日期。否则可能会收到缓存结果，且操作可能不会触发连接。

#### 6.2.11 尝试在 Linux VM 上装载 Azure 文件时出现“装入错误 115”

##### 原因

Linux 分发尚不支持 SMB 3.0 中的加密功能。在某些分发中，用户尝试通过 SMB 3.0 装载 Azure 文件时可能因功能缺失而收到“115”错误消息。

##### 解决方案

如果使用的 Linux SMB 客户端不支持加密，装载 Azure 文件时使用的 SMB 2.1 需来自文件存储账户所在的数据中心内的 Linux VM。

#### 6.2.12 Linux VM 在类似“ls”的命令中遇到随机延迟

##### 原因

装载命令中没有 **serverino** 选项时，会发生此情况。若没有 **serverino**，ls 命令会在每个文件上运行 **stat**。

##### 解决方案

请检查“/etc/fstab”条目中的 **serverino**：

```
//azureuser.file.core.chinacloudapi.cn/wms/comer on /home/sampledირ type  
cifs (rw,nodev,relatime,vers=2.1,sec=ntlmssp,cache=strict,username=xxx,  
domain=X, file_mode=0755,dir_mode=0755,serverino,rsize=65536,wsiz=65536,  
actimeo=1)
```

如果 **serverino** 选项不存在，请选中 **serverino** 选项，卸载并再次装载 Azure 文件。

#### 6.2.13 错误 112 - 超时错误

此错误指示出现通信故障，导致在使用“软”装载选项（默认设置）时无法重新与服务器建立 TCP 连接。

##### 原因

此错误的原因可能是出现 Linux 重新连接问题，或者存在其他阻止重新连接的问题，例如网络错误。指定硬装载会强制客户端等到建立连接或者显式中断为止，可用于避免由于网络超时而引起的错误。但用户应注意，这可能会导致无限期等待，应在必要时停止连接。

##### 解决方法

Linux 问题已得到修复，但更新尚未移植到 Linux 分发版。如果此问题是由 Linux 中的

重新连接问题造成的，可以通过避免进入空闲状态来解决。若要实现此目的，可在 Azure 文件共享中保留一个文件并每隔 30 秒或更短时间向其写入数据。此操作必须为写入操作，例如在文件上重新写入创建/修改日期。否则可能会收到缓存结果，且操作可能不会触发连接。

#### 6.2.14 从其他应用程序访问

是否可以通过 Web 作业引用应用程序的 Azure 文件共享？

无法在 appservice 沙盒中装载 SMB 共享。一种可能的解决方法是将 Azure 文件共享映射为映射驱动器，并允许应用程序以驱动器号的形式访问它。

### 6.3 普通存储

#### 6.3.1 普通存储概述

Azure 存储是依赖于持续性、可用性和可缩放性来满足用户需求的现代应用程序的云存储解决方案。目前普通存储账号提供四种服务：Blob 存储、表、队列消息和文件服务。

**Blob 存储：**用于存放非结构化数据，Blob 可以是任何类型的文本或二进制数据，Blob 存储分为页 Blob 和块 Blob，虚拟机的磁盘只能使用页 Blob。

**表：**通常用于存放结构化数据，表中数据主要以 NoSQL 键值对的形式存在，可以用于实现快速开发以及数据的快速访问。

**队列消息：**可以为云服务各个组件之间通信以及工作流提供可靠的消息传递和缓存。

**文件服务：**使用标准 SMB 协议，为应用程序提供文件目录共享服务。

对于 Blob 存储服务来说，Azure 提供了“容器”用于对 Blob 文件进行组织，对于具有相同用途的 Blob 文件，可以将其放置在同一个容器中，每个 Blob 文件都被包含在某个容器中，一个容器可以存放的 Blob 文件数量不限，直到达到普通存储账号 500TB 的容量上限。默认情况下，Azure 会在虚拟机所使用的存储账号中创建一个名为“vhds”的容器，用于存放虚拟机的系统磁盘和数据磁盘对应的 VHD 文件。

除了用于 Blob 文件归类，容器还提供了三种不同的访问策略（私有，公共容器，公共 Blob）用于对容器内的数据进行保护：

**私有：**默认的容器访问策略，仅允许存储账户的所有者访问容器中的内容，所有访问需要携带存储账号的管理访问密钥。

**公共容器：**允许对容器的全部元数据及容器中 Blob 文件的访问。

**公共 Blob：**仅允许对容器中的 Blob 文件进行访问，不允许访问容器的元数据。

对于选定容器，可以在经典管理界面中调整其访问策略，如图 6.3-1 所示。

普通存储账号的管理访问密钥可以在存储账号仪表板下方按钮中找到，如图 6.3-2 所示。





图 6.3-1



图 6.3-2

6.3.2 性能

普通存储中，单个 Blob 文件的 IOPS 上限为 500，当 IOPS 的瞬时值高于 500 时，会出现延迟的 I/O 请求，每个普通存储账户的 IOPS 上限为 20000，当瞬时的总 IOPS 高于 20000 时，也会出现延迟的 I/O 请求。

因此，为了避免由于延迟的 I/O 请求导致数据不一致、文件系统异常等问题，Azure 本身存在相应的保护机制。同时，用户也应当合理分配 I/O 资源，针对 I/O 资源需求比较高的应用，使用磁盘阵列或者高级存储来减少延迟的 I/O 数量。

对于使用普通存储的虚拟机，可以根据虚拟机的型号来合理规划其中存放的磁盘文件数量。例如对于基本层的虚拟机，磁盘的 IOPS 上限为 300，所以如果全部磁盘的使用均达到上限，则最多不应该在同一个存储账号下同时存放超过 66 块（20000/300）磁盘；对于标准层的虚拟机，则应该尽量避免在同一存储账号下存放超过 40 块（20000/500）磁盘。当然这两个例子中仅仅是理论值，对于 I/O 密集的应用而言，应当尽量将其磁盘放在独立的存储账号下，从而避免由于延迟 I/O 引发的各种问题。

6.3.3 使用 Raid 和 LVM 提升磁盘性能并实现磁盘动态扩展

本节提供了一个通过 Raid 来提升磁盘性能，并通过 LVM 实现磁盘大小动态扩展的配置步骤参考。

最终分区结构见表 6.3-1。

表 6.3-1

逻辑卷	/dev/mapper/VolGroup1-LogicalVol1				/dev/mapper/VolGroup1-LogicalVol2			
逻辑卷组	/dev/VolGroup1							
RAID	/dev/md127			/dev/md126			/dev/md125	
分区	/dev/sdc1	/dev/sdc2	/dev/sdd1	/dev/sdd2	/dev/sde1	/dev/sde2	/dev/sdf1	/dev/sdf2
磁盘	/dev/sdc		/dev/sdd		/dev/sde		/dev/sdf	

具体的配置步骤如下：

(1) 磁盘分区。

首先，创建一台 CentOS 6.7 的虚拟机，并挂载 4 块 10GB 的数据磁盘：

```
[root@CentOS67 testuser]# fdisk -l
.....
Disk /dev/sdc: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

.....
Disk /dev/sdf: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

接着分别将每个磁盘分为 2 个分区，以 /dev/sdc 为例：

使用 fdisk 工具划出第一个分区 /dev/sdc1，大小约为 5GB（dev/sdc2 同理，Partition number 要改为 2，扇区的起止编号也不同，sdc2 扇区编号是 654-1305）：

```
[root@CentOS67 testuser]# fdisk /dev/sdc

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default 1305): 653

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
```

```
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

使用 `fdisk -l` 查看分区结果:

```
[root@CentOS67 testuser]# fdisk -l

.....
Disk /dev/sdc: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xbd293e5b
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	653	5245191	fd	Linux raid autodetect
/dev/sdc2		654	1305	5237190	fd	Linux raid autodetect

对剩下的 3 个磁盘 (`/dev/sdd`, `/dev/sde`, `/dev/sdf`) 进行同样的处理, 处理后使用 `fdisk -l` 查看是否全部完成分区格式化。

(2) RAID 5 的配置。

采用 RAID 5, 在提高 IO 性能的同时保证数据安全。

首先加载 `raid5` 内核模块:

```
[root@CentOS67 testuser]# modprobe raid5
```

接下来将 `/dev/sdc1`, `/dev/sdc2`, `/dev/sdd1` 合并为 `/dev/md127`:

```
[root@CentOS67 testuser]# mdadm --create /dev/md127 --level=5
--raid-devices=3 /dev/sd[cd]1 /dev/sdc2
mdadm: /dev/sdc1 appears to contain an ext2fs file system
size=5245188K mtime=Thu Jan 1 00:00:00 1970
mdadm: /dev/sdd1 appears to contain an ext2fs file system
size=5245188K mtime=Thu Jan 1 00:00:00 1970
mdadm: /dev/sdc2 appears to contain an ext2fs file system
size=5237188K mtime=Thu Jan 1 00:00:00 1970
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md127 started.
```

查看 `/dev/md127` 的状态:

```
[root@CentOS67 testuser]# mdadm --misc --detail /dev/md127
```

类似地, 分别将 `/dev/sdd2`, `/dev/sde1`, `/dev/sde2` 合并为 `/dev/md126`, 将 `/dev/sdf1` 和 `/dev/sdf2` 合并为 `/dev/md125`。

## LVM 配置:

首先, 将/dev/md127, /dev/md126, /dev/md125 创建为 Physical Volume(PV):

```
[root@CentOS67 testuser]# pvcreate /dev/md127 /dev/md126 /dev/md125
Physical volume "/dev/md127" successfully created
Physical volume "/dev/md126" successfully created
Physical volume "/dev/md125" successfully created
```

扫描 Physical Volume 的改动:

```
[root@CentOS67 testuser]# pvscan
PV /dev/md125          lvm2 [4.99 GiB]
PV /dev/md126          lvm2 [9.98 GiB]
PV /dev/md127          lvm2 [9.98 GiB]
Total: 3 [24.95 GiB] / in use: 0 [0   ] / in no VG: 3 [24.95 GiB]
```

将 /dev/md127 和 /dev/md126 添加到 Volume Group 中:

```
[root@CentOS67 testuser]# vgcreate VolGroup1 /dev/md127 /dev/md126
Volume group "VolGroup1" successfully created
```

Volume Group 创建好后, 在其上划分新的 Logical Volume:

```
[root@CentOS67 testuser]# lvcreate -l 2500 -n LogicalVol1 VolGroup1
Logical volume "LogicalVol1" created.
[root@CentOS67 testuser]# lvcreate -l 2608 -n LogicalVol2 VolGroup1
Logical volume "LogicalVol2" created.
```

使用 mkfs.ext4 工具将两个 Logical Volume 的文件系统格式化为 ext4:

```
[root@CentOS67 testuser]# mkfs.ext4 /dev/mapper/VolGroup1-LogicalVol1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
640848 inodes, 2560000 blocks
128000 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2621440000
79 block groups
32768 blocks per group, 32768 fragments per group
8112 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 39 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[root@CentOS67 testuser]# mkfs.ext4 /dev/mapper/VolGroup1-LogicalVol2
.....
```

格式化后，将格式化得到的文件分区 **mount** 到目录中（如果要开机自动挂载可以修改 `/etc/fstab`）：

```
[root@CentOS67 testuser]# mkdir /mnt/LV1
[root@CentOS67 testuser]# mkdir /mnt/LV2
[root@CentOS67 testuser]# mount /dev/mapper/VolGroup1-LogicalVol1 /mnt/LV1
[root@CentOS67 testuser]# mount /dev/mapper/VolGroup1-LogicalVol2 /mnt/LV2
```

至此，LVM 部分已经配置完成。

(3) 进一步测试。

扩展 LVM：

下载一个测试文件，放到 LVM2 分区中：

```
[root@CentOS67 LV2]# wget
http://daneaststorage.blob.core.chinacloudapi.cn/demo/Azure.pdf
--2017-03-09 15:13:21--
http://daneaststorage.blob.core.chinacloudapi.cn/demo/Azure.pdf
Resolving daneaststorage.blob.core.chinacloudapi.cn... 42.159.208.78
Connecting to daneaststorage.blob.core.chinacloudapi.cn|42.159.208.78|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7670041 (7.3M) [application/pdf]
Saving to: "Azure.pdf"

100%[=====] 7,670,041
-----K/s in 0.06s

2017-03-09 15:13:21 (123 MB/s) - "Azure.pdf" saved [7670041/7670041]

[root@CentOS67 LV2]# ll
total 7508
-rw-r--r--. 1 root root 7670041 Jul 6 2016 Azure.pdf
drwx-----. 2 root root 16384 Mar 9 15:02 lost+found
```

将创建好的 Physical Volume `/dev/md125` 扩展到 VolGroup1 中：

```
[root@CentOS67 LV2]# vgextend VolGroup1 /dev/md125
Volume group "VolGroup1" successfully extended
```

扩展 Logic Volume 的大小：

```
[root@CentOS67 LV2]# lvextend /dev/mapper/VolGroup1-LogicalVol2 /dev/md125
Size of logical volume VolGroup1/LogicalVol2 changed from 10.19 GiB (2608
extents) to 15.18 GiB (3885 extents).
Logical volume LogicalVol2 successfully resized
```

扩展完成后，可以看到 /mnt/LV2 的大小实际没有发生变化：

```
[root@CentOS67 LV2]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdal        30G  2.0G   26G   7% /
tmpfs            6.9G   0   6.9G   0% /dev/shm
/dev/sdb1        133G   60M  126G   1% /mnt/resource
/dev/mapper/VolGroup1-LogicalVol1
                 9.5G   22M   9.0G   1% /mnt/LV1
/dev/mapper/VolGroup1-LogicalVol2
                 10G   33M   9.4G   1% /mnt/LV2
```

需要进一步使用 `resize2fs` 工具将文件系统的大小进行扩展：

```
[root@CentOS67 LV2]# resize2fs /dev/mapper/VolGroup1-LogicalVol2
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/VolGroup1-LogicalVol2 is mounted on /mnt/LV2;
on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/mapper/VolGroup1-LogicalVol2 to 3978240
(4k) blocks.
The filesystem on /dev/mapper/VolGroup1-LogicalVol2 is now 3978240 blocks
long.
```

查看发现文件系统的大小已经变为新的大小（扩展成功）：

```
[root@CentOS67 LV2]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdal        30G  2.0G   26G   7% /
tmpfs            6.9G   0   6.9G   0% /dev/shm
/dev/sdb1        133G   60M  126G   1% /mnt/resource
/dev/mapper/VolGroup1-LogicalVol1
                 9.5G   22M   9.0G   1% /mnt/LV1
/dev/mapper/VolGroup1-LogicalVol2
                 15G   33M   15G   1% /mnt/LV2
```

文件未损坏或丢失：

```
[root@CentOS67 LV2]# ll
total 7508
-rw-r--r--. 1 root root 7670041 Jul  6 2016 Azure.pdf
drwx-----. 2 root root  16384 Mar  9 15:02 lost+found
```

## 6.4 高级存储

### 6.4.1 高级存储概述

针对对于磁盘读写性能有更高要求的应用程序，Azure 提供了高级存储账号来满足其需求，高级存储提供了高性能、低延迟的存储服务。

目前在经典管理门户上还无法通过图形界面对高级存储进行创建和管理，可以使用 Azure Powershell 对高级存储账号进行创建和配置，也可以使用新版本管理门户进行管理配置。对于高级存储账号，目前仅支持页 Blob（Page Blob），即高级存储目前主要用于持久化虚拟机磁盘中的数据。

要使用高级存储账号，需要特定型号的虚拟机，目前支持挂载高级存储账号中的磁盘的虚拟机型号主要有 DS 系列、FS 系列、DSv2 系列这类名称中带“S”的机型。目前经典门户中不支持创建此类虚拟机，要在经典模式下创建这类虚拟机可以使用 Azure Powershell 来操作，或者可以使用新版本管理门户进行创建。要在资源管理器模式中创建此类虚拟机，可以直接通过新版本管理门户完成操作。对于此类型号的虚拟机而言，并非只可以挂载高级存储账号中的磁盘，可以在其上同时挂载普通存储和高级存储的磁盘。

对于已经在普通存储账号中创建的虚拟机，如果磁盘性能已经无法满足应用的需要，可以通过将其磁盘迁移至高级存储账号下，并将虚拟机升级为可挂载高级存储磁盘的机型，升级过程由于涉及到磁盘从普通存储账号到高级存储账号的迁移操作，所以需要对虚拟机的重建操作。在创建虚拟机的时候，应该尽量避免将普通型号的虚拟机（例如 A，D，F 系列）与可挂载高级存储磁盘的虚拟机（例如 DS，FS 系列）放在同一个云服务下。

## 6.4.2 性能

高级存储账号下创建的磁盘随着大小不同有不同的性能指标，同时其价格也有所差异（高级存储账号中的磁盘的计费模式也有所不同），目前有三种类型的高级存储磁盘，性能如表 6.4-1 所示。

表 6.4-1 高级存储磁盘型号表

磁盘型号	P10	P20	P30
磁盘大小	128GB	512GB	1023GB
单块磁盘 IOPS	500	2300	5000
单块磁盘吞吐量	100MB/s	150MB/s	200MB/s

对于高级存储账号中的磁盘并非只能限制为上述表格中的大小，当磁盘大小在上面两个型号中间时，会将磁盘型号映射到所在区间较高的一个型号，例如当挂载一个 400GB 的磁盘时，处于 P10 和 P20 之间，所以磁盘类型为 P20，性能也满足 P20 的性能，如果磁盘大小为 513GB，处于 P20 和 P30 之间，则磁盘类型符合 P30，性能也与 P30 型号的磁盘一致。

磁盘的型号随磁盘大小改变而动态改变，当磁盘需要调整磁盘性能时，可以通过 Azure Powershell 命令或者新版本管理门户对磁盘进行大小缩放，从而实现对其性能的调整。

对于表格中的 IOPS 指标，单次 I/O 的大小限制为 256KB。如果单次 I/O 小于 256KB，则视为 1 次 I/O 操作，如果大于 256KB，则将其视为大小为 256KB 的多次 I/O。

表格中的吞吐量是包括磁盘写入和读取（非缓存数据）的总和，即对于 P10 而言，每秒读写的吞吐量之和最高不会超过 100MB。

与普通存储账号相比，高级存储账号不存在 20000 IOPS 的限制因素。

由于 Azure 中存储与计算节点分离的设计，所以制约虚拟机最终磁盘性能的因素并非只有磁盘的 IOPS 和吞吐量本身，计算节点（即虚拟机）与存储账号之间的“带宽”指标也是考虑因素之一，关于支持高级存储的不同虚拟机型号对应的存储带宽指标，请参考计算与云服务一章中关于不同型号虚拟机非缓存磁盘最大吞吐量的指标。只有当虚拟机的非缓存磁盘最大吞吐量指标高于其挂载的磁盘的 IOPS 和吞吐量指标时，才能够最大程度地发挥磁盘的性能。例如：对于 Standard\_DS2 型号的虚拟机，其非缓存磁盘最大吞吐量指标为 6400 IOPS 和 64MB/s，如果将一块 P30 的磁盘挂载到这台虚拟机，虽然 P30 磁盘的单块吞吐量为 200MB/s，由于虚拟机型号限制，最大也只能达到 64MB/s 的性能。

对于非 I/O 密集的磁盘来说，可以在成本与性能之间进行进一步优化和考虑，例如如果应用的 I/O 主要分布在数据磁盘中，则可以将虚拟机的系统磁盘部署到普通存储账号下（即数据磁盘使用高级存储账号，系统磁盘放置在普通存储账号中），从而实现成本节约。

对于 Linux 虚拟机而言，想要充分发挥高级存储账号的性能，还有一些注意事项：

（1）对于缓存规则配置为“ReadOnly”或“None”的高级存储磁盘而言，需要在挂载文件系统时禁用“barrier”，以实现更高性能，对于这类缓存规则的磁盘来说，磁盘的“写”操作并不缓存，所以写操作完成后，数据就已经被存入存储中，所以此时启用“barrier”反而会对性能有影响。对于 reiserFS 文件系统来说，在挂载时通过指定参数“barrier=none”来禁用“barrier”，对于 ext3 和 ext4 来说，使用参数“barrier=0”，对于 XFS 来说，则使用参数“nobarrier”。

（2）对于缓存规则配置为“ReadWrite”的高级存储磁盘而言，需要启用“barrier”来避免文件系统崩溃。

对于下面表格中的 Linux 发行版以及内核版本来说，能够更好地发挥高级存储的性能优势，同时某些版本还需要配合更高版本的驱动来实现性能优化见表 6.4-2。

表 6.4-2

分 发	版 本	支持的内核	支持的映像
Ubuntu	12.04	3.2.0-75.110	Ubuntu-12_04_5-LTS-amd64-server-20150119-zh-CN-30GB
	14.04	3.13.0-44.73	Ubuntu-14_04_1-LTS-amd64-server-20150123-zh-CN-30GB
	14.10	3.16.0-29.39	Ubuntu-14_10-amd64-server-20150202-zh-CN-30GB
	15.04	3.19.0-15	Ubuntu-15_04-amd64-server-20150422-zh-CN-30GB
SUSE	SLES 12	3.12.36-38.1	suse-sles-12-priority-v20150213 suse-sles-12-v20150213
CoreOS	584.0.0	3.18.4	CoreOS 584.0.0
CentOS	6.5, 6.6, 6.7, 7.0		需要 LIS 4.0
	7.1	3.10.0-229.1.2.el7	建议使用 LIS 4.0
Oracle	6.4		需要 LIS 4.0
	7.0		

如果用户使用高级存储磁盘的虚拟机的内核版本后者驱动版本低于上面表格中的推荐版本，建议进行版本升级以充分发挥高级存储的性能。



### 6.4.3 使用 Azure Powershell 创建 DS 系列虚拟机并附加磁盘

安装步骤这里不再赘述，打开 Azure Powershell 并导入订阅信息后，使用下面的命令创建高级存储账号：

```
New-AzureStorageAccount -StorageAccountName "<高级存储账号名称>" -Location
"<China East/China North>" -Type "Premium_LRS"
```

创建成功后，使用下面的命令设置当前订阅对应的默认存储账号为上面创建的高级存储：

```
Set-AzureSubscription -SubscriptionName "<订阅名称>" -CurrentStorage
AccountName "<高级存储账号名称>"
```

完成后，使用下面的命令创建一台 Standard\_DS2 的虚拟机，命令中的参数需要根据实际情况进行修改，这里仅仅举例说明：

```
$storageAccount = "<高级存储账号名称>"
$adminName = "<管理员用户名>"
$adminPassword = "<管理员密码>"
$vmName = "<虚拟机名称>"
$location = "<China East/China North>"
#可以使用 Get-AzureVMImage 来查看需要的 imageName，也可以使用自定义的映像
#这里以 2012 datacenter 为例，注意映像日期的变化，可能会随着平台更新而失效
$imageName = "55bc2b193643443bb879a78bda516fc8__Windows-Server-2012-Datacenter-201504.01-zh.cn-127GB.vhd"
$vmSize = "Standard_DS2"
$OSDiskPath = "https://" + $storageAccount + ".blob.core.chinacloudapi.cn/vhds/" + $vmName + "_OS_PIO.vhd"
$vm = New-AzureVMConfig -Name $vmName -ImageName $imageName -InstanceSize $vmSize -MediaLocation $OSDiskPath
#windows 使用这条命令
Add-AzureProvisioningConfig -Windows -VM $vm -AdminUsername $adminName -Password $adminPassword
#如果是 Linux 虚拟机，使用下面的命令：
#Add-AzureProvisioningConfig -Linux -VM $vm -LinuxUser $adminName -Password $adminPassword
New-AzureVM -ServiceName $vmName -VM $vm -Location $location
```

上面的命令中仅包含了虚拟机的基本配置，如果可以实现更多的配置细节，可以参考 Azure 官网关于使用 Powershell 创建虚拟机的对应说明。

对于创建好的虚拟机，可以使用下面的命令为虚拟机附加高级存储的数据磁盘：

```
$storageAccount = "<高级存储账号名称>"
$vmName = "<虚拟机名称>"
$vm = Get-AzureVM -ServiceName $vmName -Name $vmName
$LunNo = "<磁盘 Lun 号>"
$path = "http://" + $storageAccount + ".blob.core.chinacloudapi.cn/vhds/"
```

```
+ "myDataDisk_" + $LunNo + "_PIO.vhd"
    $label = "Disk " + $LunNo
    Add-AzureDataDisk -CreateNew -MediaLocation $path -DiskSizeInGB <磁盘大小>
-DiskLabel $label -LUN $LunNo -HostCaching <ReadWrite/ReadOnly/None> -VM $vm
| Update-AzureVm
```

当然，也完全可以通过经典管理界面或者新版本的管理界面为创建好的虚拟机附加高级存储磁盘。

#### 6.4.4 将虚拟机磁盘从普通存储迁移到高级存储

目前 Azure 平台不支持将普通型号的虚拟机（例如 A, D, F 系列）直接通过调整大小的方式升级为可使用高级存储的虚拟机（例如 DS, FS 系列），因此需要首先以“保留附加的磁盘”的方式将需要迁移的虚拟机删除（对于资源管理器模型下的虚拟机，“保留磁盘”是默认的删除方式），如图 6.4-1 所示。



图 6.4-1

删除后，虚拟机的系统磁盘和所有的数据磁盘会保留下来，接着需要通过 Azure Powershell 或存储管理工具将这些磁盘对应的 VHD 文件拷贝到已有的高级存储账号下。

拷贝完成后，通过下面的 Azure Powershell 命令（经典模型）将 VHD 分别创建为系统磁盘和数据磁盘：

```
#系统磁盘:
Add-AzureDisk -DiskName "<磁盘名称>" -MediaLocation "<系统磁盘 VHD 的 URL>"
-OS "<Windows/Linux>"
#数据磁盘:
Add-AzureDisk -DiskName "<磁盘名称>" -MediaLocation "<数据磁盘 VHD 的 URL>"
```

完成后，使用创建好的系统磁盘创建虚拟机，虚拟机型号可以选择例如 DS, FS 系列的虚拟机。完成创建后，再将创建好的数据磁盘挂载到虚拟机上即可。

## 6.5 存储管理工具

### 6.5.1 Microsoft Azure Storage Explorer

安装完成后，首先配置 Storage Account，如图 6.5-1 所示。

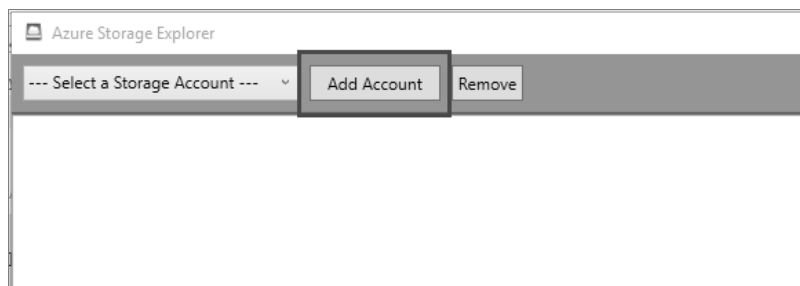


图 6.5-1

填写相关的信息，如图 6.5-2 所示。

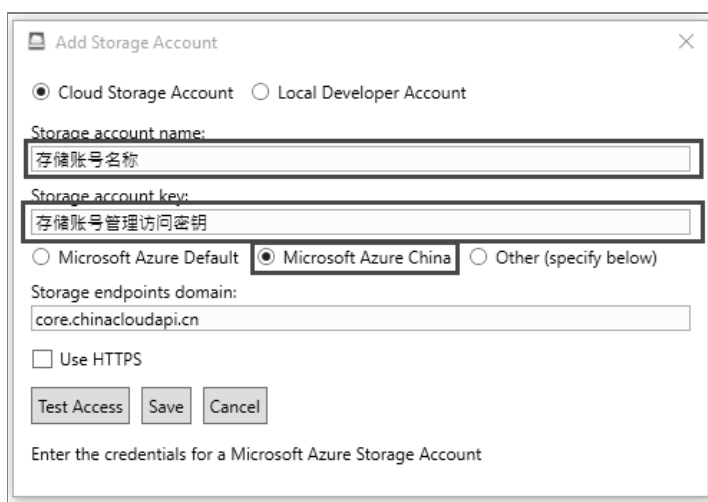


图 6.5-2

存储账号名称，如图 6.5-3 所示。

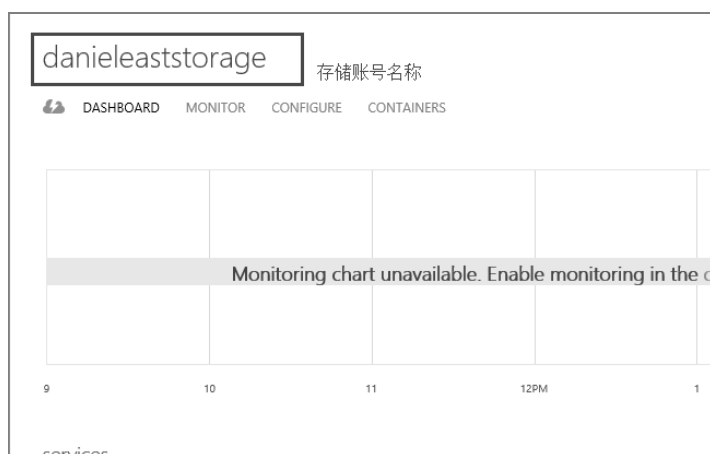


图 6.5-3

存储账号管理访问密钥，如图 6.5-4 和图 6.5-5 所示。

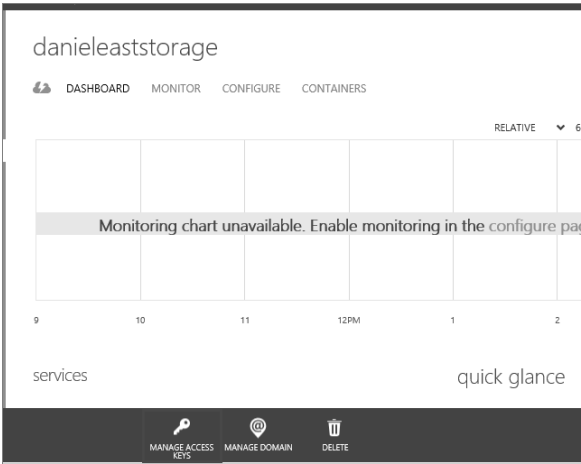


图 6.5-4

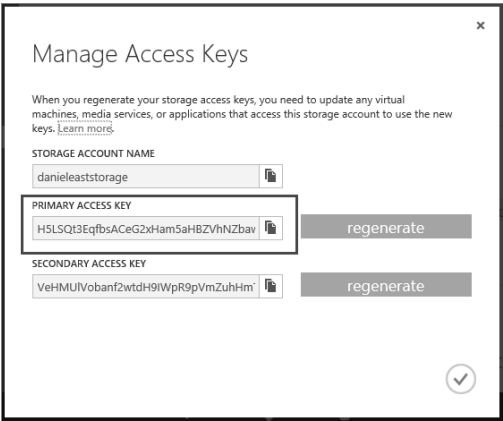


图 6.5-5

添加完成后就可以看到存储账号下的容器和文件了，如图 6.5-6 所示。

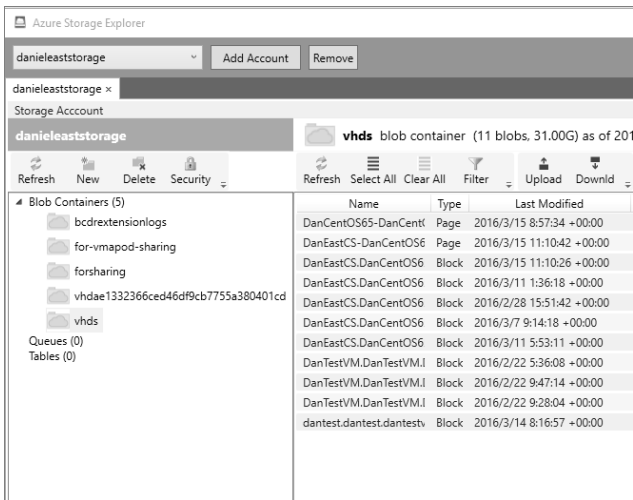


图 6.5-6

单击  新建一个容器，选择 Public Container，如图 6.5-7 所示。

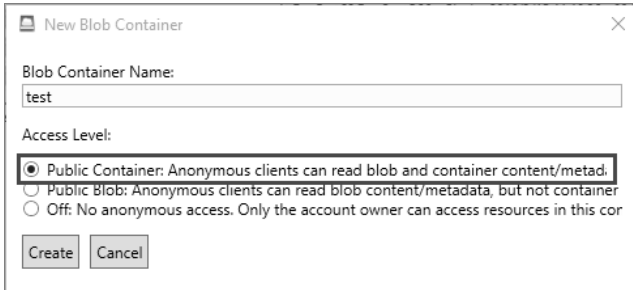



图 6.5-7

创建完成后，选中 test 容器，然后单击  将文件上传到该容器中。

## 6.5.2 CloudBerry Explorer

CloudBerry Explorer 下载地址: <http://www.cloudberrylab.com/free-microsoft-azure-explorer.aspx>

下载后简单安装后就可以开始使用，首先要完成账号的添加工作，填写 Account 和 Shared key，同时指定 Account type 为“Azure in China”，如图 6.5-8 和图 6.5-9 所示。

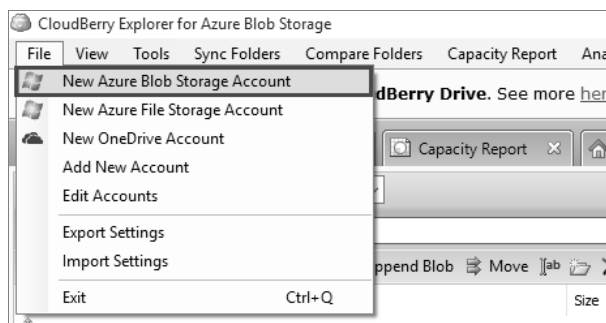


图 6.5-8



图 6.5-9

Account 和 Shared key 可以在下面这里查找到。

Account，如图 6.5-10 所示。

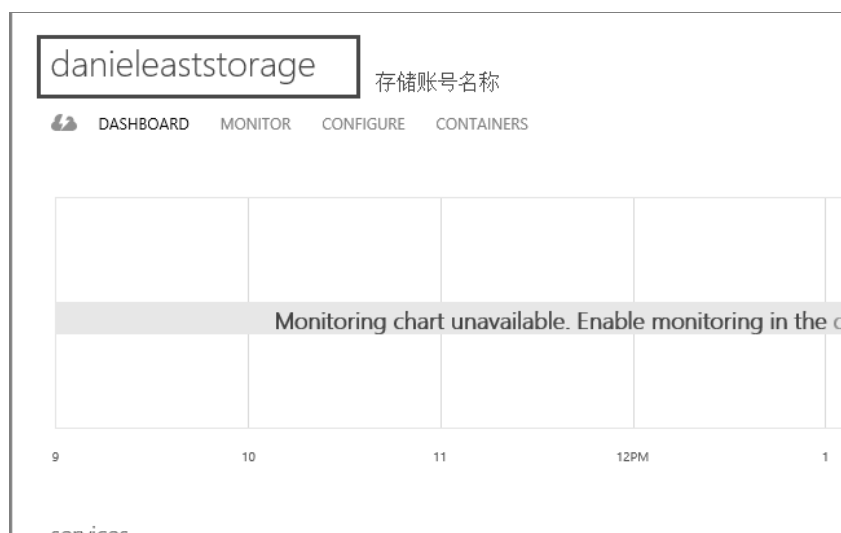


图 6.5-10

Shared key，如图 6.5-11 和图 6.5-12 所示。

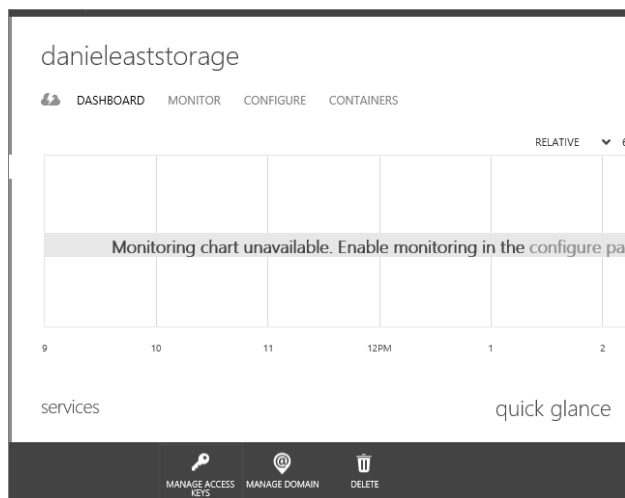


图 6.5-11

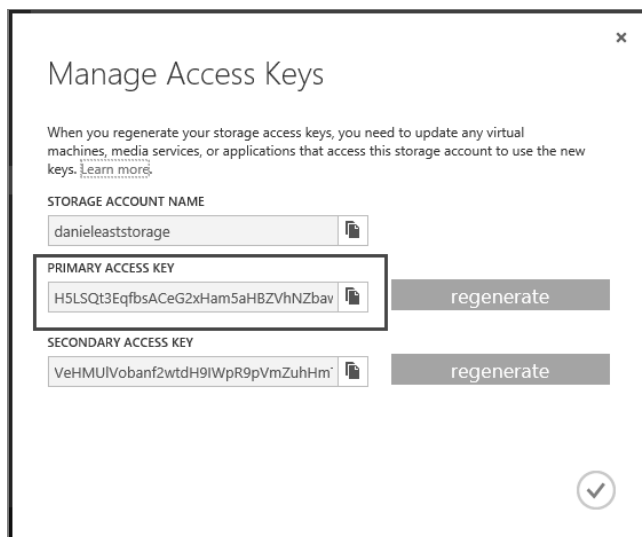


图 6.5-12

添加完成后，可以在 Source 中选择对应的存储账号，管理里面的资源，如图 6.5-13 所示。

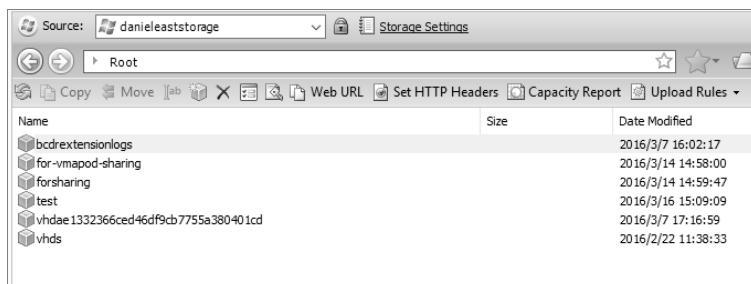


图 6.5-13

在左右两边可以选择不同的 Source，以实现不同存储账号之间的文件拷贝等操作，如图 6.5-14 所示。

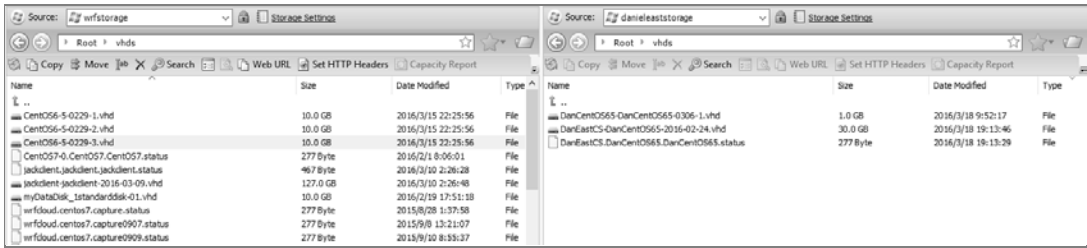


图 6.5-14

同时也可以左侧或者右侧的 Source 中选择本地磁盘，以实现上传和下载 Blob 文件。在传输（上传或下载）的时候，下方可以看到上传的速度，进度，预计剩余时间等信息，非常人性化，如图 6.5-15 所示。

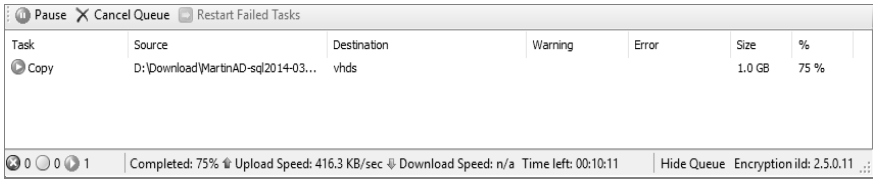


图 6.5-15

需要注意的一点是，由于 vhd 创建磁盘使用的时候，要求为 Page Blob 类型，而默认在两边拖拽的话，会以 Block Blob 的形式进行传输，所以如果是 vhd 文件，请务必使用下边的“Copy as Page Blob”功能，而不要直接拖拽，如图 6.5-16 所示。

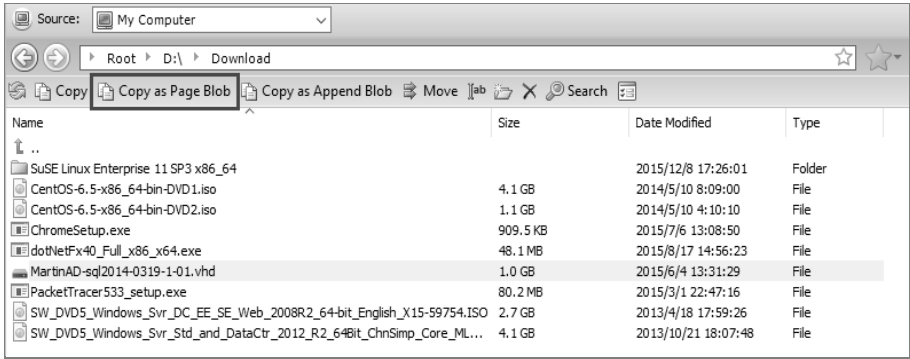


图 6.5-16

额外介绍一个 CloudBerry Explorer 功能，Capacity Report 功能，可以用来查看存储账号下的容器的容量使用情况，如图 6.5-17 所示。

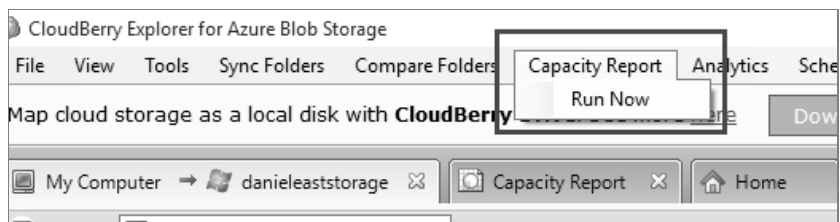


图 6.5-17

选择 Run Now 后，会进入到 Capacity Report 的标签页，如图 6.5-18 所示。

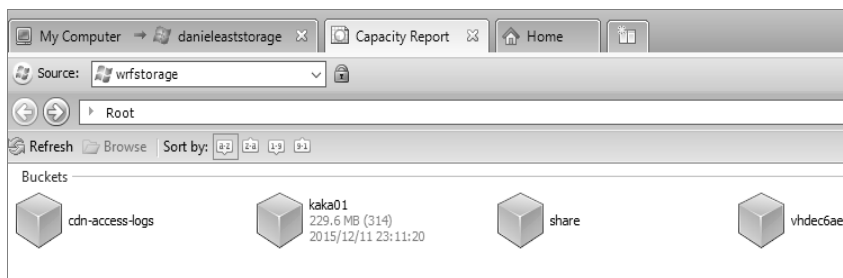


图 6.5-18

里面可以看到所有的 Container 以及使用情况，单击进去，可以进一步看到里面的 SubFolder，如图 6.5-19 所示。

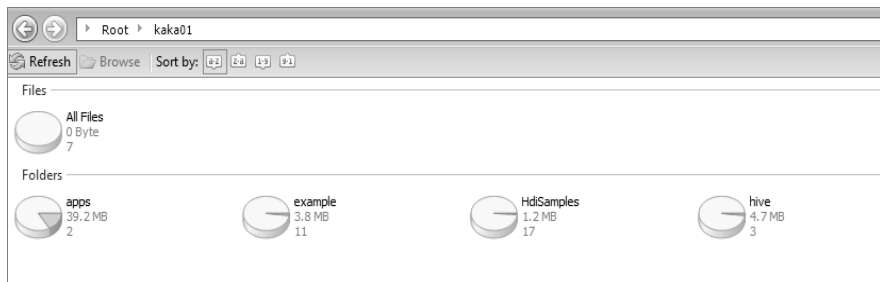


图 6.5-19

可以展开每个层级，如图 6.5-20 所示。



图 6.5-20



这个功能里面统计了 Container 和 SubFolder 的使用量（总的实际大小），而且有清晰的层级关系。

目前这个工具不可以用来打破 vhd 的 lease, 要 Break lease, 还只能使用 Azure Explorer:

<http://www.cerebrata.com/products/azure-explorer/introduction>

### 6.5.3 Azcopy 已知问题和最佳实践

#### 6.5.3.1 限制复制数据时的并发写入

在使用 AzCopy 复制 blob 或文件时, 请记住, 在复制数据时其他应用程序可能正在修改该数据。如果可能, 请确保要复制的数据在复制操作期间不会被修改。例如, 当复制与 Azure 虚拟机关联的 VHD 时, 请确保当前没有其他应用程序正在向该 VHD 进行写入。执行此操作的一个好方法是租用要复制的资源。另外, 还可以先创建 VHD 的快照, 然后复制该快照。

如果在复制 blob 或文件时无法阻止其他应用程序向其进行写入, 请记住, 在作业完成时, 复制的资源可能不再与源资源完全相同。

#### 6.5.3.2 在一台计算机上运行一个 AzCopy 实例。

AzCopy 旨在最大程度上利用计算机资源来加快数据传输, 如果需要更多的并发操作, 我们建议在一台计算机上只运行一个 AzCopy 实例并指定选项 /NC。有关详细信息, 请在命令行中键入 AzCopy /?:NC。

6.5.3.3 当进行“使用适用于加密、哈希和签名的 FIPS 兼容算法”时, 请启用适用于 AzCopy、与 FIPS 兼容的 MD5 算法。

默认情况下, 在复制对象时, 如有需要 AzCopy 启动 FIPS 兼容的 MD5 设置的某些安全需求时, AzCopy 则会使用 .NET MD5 实现来计算 MD5。

可以创建属性为 AzureStorageUseV1MD5 的 app.config 文件 AzCopy.exe.config, 并将其与 AzCopy.exe 分开放。

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="AzureStorageUseV1MD5" value="false" />
  </appSettings>
</configuration>
```

当属性“AzureStorageUseV1MD5”为 True（默认值）时, AzCopy 将使用 .NET MD5 实现; 为 False 时, AzCopy 将使用兼容 FIPS 的 MD5 算法。

请注意, 默认情况下, Windows 计算机上禁用 FIPS 兼容的算法, 可以在运行的窗口中键入 secpol.msc 并在“安全设置”→“本地策略”→“安全选项”→“系统加密”处检查此开关: 使用 FIPS 兼容算法来加密、哈希和签名。

# 第七章 网 络

除了计算和存储服务，IaaS 最重要的一个服务便是网络服务了，本章包含了丰富的网络实验内容以及案例解析，深入介绍了 Azure 虚拟网络中 IP 地址的相关概念，用户定义路由，子网划分，虚拟网络网关，点到站点 VPN，站点到站点 VPN，BGP VPN，以及能够提供更稳定的访问速度的专线服务等。

## 7.1 IP 地址相关

### 7.1.1 保留 IP 地址

保留 IP 地址这一概念是针对经典环境部署的虚拟机的公网 IP 资源而言的。Azure 平台 IaaS 虚拟机的 IP 配置如图 7.1-1，默认情况下 Azure 平台分配的公共 IP 地址默认为动态 IP 地址，且会在关闭资源或释放资源的情况下发生变化。这样的公网 IP 不适用于需要固定公网 IP 地址的服务（例如：需要 IP 地址做备案的 Web 服务）。

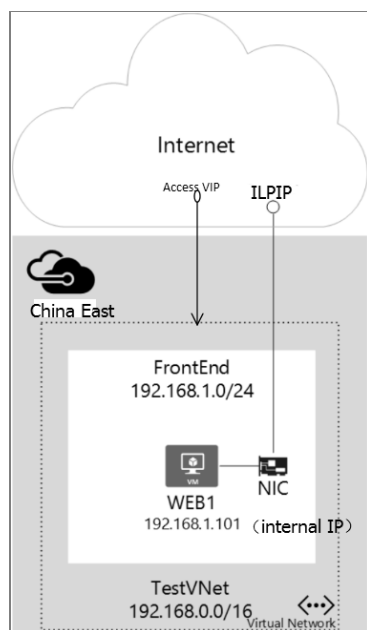


图 7.1-1 IaaS 虚拟机的 IP 配置

Azure 平台提供了 Azure PowerShell 命令行接口，为用户做保留 IP 地址。用户可以将虚拟机当前正在使用的动态 VIP 地址设置为保留 IP，设置为保留 IP 后，该 IP 地址不会因虚拟机关闭或释放资源而变化。保留 IP 会始终与你的订阅相关联，直至主动将它删除。

#### 7.1.1.1 Azure powershell 操作为 IaaS 虚拟机设置保留 IP 地址

需要实现的目标如下：

- 新建虚拟机并使用保留 IP 地址
- 即使在云服务中的 VM 处于关闭或停止（释放）状态下也不会更改的公网 IP 地址。
- 适用于在区域虚拟网络中的虚拟机

操作步骤：

- (1) 安装 Windows Azure Powershell 模块并关联 Azure 订阅账户。
- (2) 创建一个 Reserved IP，不能具体指定某个 IP 地址，Azure 平台会随机分配一个公网 IP 地址给到用户的订阅账户。

命令行输入：

```
$ReservedIP = New-AzureReservedIP -ReservedIPName <ReserveIP 自定义名称>
-Location "China North/China East "
```

使用如下命令查询新建的保留 IP：

```
Get-AzureReservedIP -ReservedIPName <ReserveIP 自定义名称> -
```

预期输出：

```
ReservedIPName : Rtest
Address        : 139.217.15.44
Id             : d73be9dd-db12-4b5e-98c8-bc62e7c42041
Label          :
Location       : China North
State          : Created
InUse          : False
ServiceName    :
DeploymentName :
OperationDescription: Get-AzureReservedIP
OperationId     : 55e4f245-82e4-9c66-9bd8-273e815ce30a
OperationStatus : Succeeded
```

#### (3) 创建新的虚拟机并使用该 ReservedIP

```
New-AzureVMConfig -Name <云服务名称> -InstanceSize <虚拟机型号> -ImageName <
使用的映像名> | Add-AzureProvisioningConfig -Windows -AdminUsername <用户名>
-Password <密码> | New-AzureVM -ServiceName <云服务名称> -ReservedIPName <Reserve
IP 名称>
```

再次查询保留 IP，可以看到该 IP 地址已经关联了云服务及虚拟机。

```

ReservedIPName    : Rtest
Address           : 139.217.15.44
Id                : 54fbe9dd-db12-4b5e-98c8-bc62e7c44561
Label             :
Location          : China North
State             : Created
InUse             : True
ServiceName       : viptest11
DeploymentName     : viptest
VirtualIPName     : viptestcontractcontract
OperationDescription : Get-AzureReservedIP
OperationId        : 2344f245-82e4-9c66-9bd8-273e815ce2wsx
OperationStatus    : Succeeded

```

至此该拥有保留 IP 地址的虚拟机已经创建完成了。为验证 `powersehl1` 命令配置的保留 IP 地址是生效的，可以尝试将云服务中的虚拟机全部关闭后再重启，看看公网 IP 地址是否会发生变化。或者用户可以登录 `azure` 新的管理界面—门户预览来查看虚拟机的 IP 地址是否显示为保留 IP。

#### 7.1.1.2 保留 IP 地址相关

- 可以为已经创建的在虚拟网络中的虚拟机配置保留 IP 地址。
- 可以将订阅中没有使用的保留 IP 地址关联到新的云服务。
- 可以使用 `powershell` 命令行删除保留 IP 地址，如果相关的云服务中有存在的虚拟机，需要先删除虚拟机才能删除 `ReservedIP`。操作命令行：

```
Remove-AzureReservedIP - ReservedIPName <ReserveIP 自定义名称>
```

- 保留 IP 地址一旦删除不能手动找回。
- `ReservedIP` 是计入收费的，具体收费需要联系 `azure` 商务部门咨询。

参考官网链接：<https://www.azure.cn/documentation/articles/virtual-networks-reserved-public-ip/>

#### 7.1.1.3 最佳实践结合

某公司将本地的某一公网 Web 服务迁移到 `Azure` 的一台虚拟机上，并在完成部署后将这台虚拟机自动获取的公网 IP 地址 `42.159.x.x` 做了 DNS A 记录。有一天发现服务无法正常访问，后经调查发现是虚拟机被误操作关机后再启动。虚拟机的公网 IP 变化了，导致无法正确找到这台服务器。

案例分析：

- 该用户部署单台虚拟机的方案本身不符合 `azure` 平台要求的同一可用性集至少 2 个及 2 个以上的实例，从而保证服务的 SLA。
- `Azure` 平台建议用户需要做 DNS 记录时，选择 C NAME 的方式绑定虚拟机对应的域名。如果必须要使用 A 记录，请将该虚拟机获取的公网 IP 地址做保留。以规避资源释放带来 IP 变化的风险。

- 如果问题已经发生，只能通过将当前获取的公网 IP 地址做保留从而避免问题再次发生。Azure 平台的技术维护也不能手动干预 IP 地址分配。已释放的 IP 地址不能手动找回。

## 7.1.2 虚拟网络静态专用 IP 地址

虚拟网络静态专用 IP 地址是指虚拟网络内的虚拟机获取的静态内网 IP 地址。默认情况下，虚拟机自动获取的内网 IP 地址是动态的，在 azure 管理界面上关闭虚拟机后再启动获取的内网 IP 地址可能会变化。这样的 IP 配置不适合需要静态内网 IP 的应用场景。（例如：虚拟机承担域 DNS 服务器 AD 服务器等角色，或者集群的配置等）

Azure 平台提供 PowerShell 命令行的方式实现为虚拟机配置静态内网 IP 地址。

### 7.1.2.1 Azure powershell 操作为 IaaS 虚拟机设置静态专用 IP 地址

需要实现的目标：

- 对已有虚拟机设置静态 Internal IP。
- 内网 IP 地址不会因为虚拟机的关闭等操作而发生变化。
- 只适用于虚拟网络中的虚拟机。

操作步骤：

- (1) 安装 Windows Azure Powershell 模块并关联 azure 订阅账户
- (2) 找到目标虚拟机

```
PS C:\> $vm = Get-AzureVM -servicename 'CloudService1' -name 'VM01'
```

- (3) 为该虚拟机指定静态内网 IP 地址并更新该虚拟机

```
PS C:\> Set-AzureStaticVNetIP -vm $vm -IPAddress 10.0.1.4 | Update-AzureVM
```

- (4) 通过以下命令查看虚拟机的静态内网 IP

```
PS C:\> Get-AzureStaticVnetIP -VM $VM
```

为验证 powershell 命令配置的静态内网 IP 地址是生效的，可以尝试将云服务中的虚拟机全部关闭后再重启，看看内网 IP 地址是否会发生变化。或者用户可以登录 azure 新的管理界面—门户预览来查看虚拟机的内网 IP 地址是否显示为静态。

### 7.1.2.2 虚拟机静态内网 IP 地址相关

- 取消虚拟机的静态 Internal IP，命令行：

```
PS C:\> $vm = Get-AzureVM -ServiceName 'CloudService1' -name 'VM01'
PS C:\> Remove-AzureStaticVNetIP $vm | Update-AzureVM
```

- 创建静态 Internal IP 的虚拟机，命令行：

```
PS C:\> $images = Get-AzureVMImage
PS C:\> $vm = New-AzureVMConfig -Name 'VM01' -ImageName $images[29].ImageName
-InInstanceSize Small |Add-AzureProvisioningConfig -Windows -AdminUsername
```

```
'testaccount' -Password 'Abcd1234' | Set-AzureSubnet -SubnetNames 'Subnet01'
| Set-AzureStaticVNetIP -IPAddress 10.0.1.10 | New-AzureVM -ServiceName
'CloudService1' -VNetName 'VNet01'
```

- 在同一虚拟网络子网中不建议混合使用静态内网 IP 地址和动态内网 IP 地址。
- 为已创建的虚拟机配置静态内网 IP 地址时虚拟机需要是运行的状态。
- 静态内网 IP 地址不需要单独付费。

参见相关链接：<http://msdn.microsoft.com/en-us/library/azure/dn630228.aspx>

### 7.1.3 实例公共 IP 地址（ILPIP）

实例公共 IP 地址是虚拟机本身的公网 IP 地址。配置了这个地址的虚拟机是直接面对公网的。图 7.1-2 说明了 IaaS 虚拟机 ILPIP 地址与 VIP 地址的区别。用户需要注意为配置了 ILPIP 地址的虚拟机做好安全防护措施。

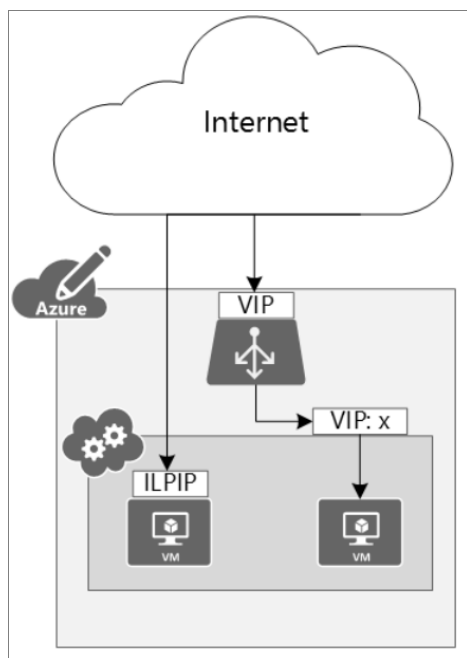


图 7.1-2

如图 7.1-2 所示，云服务是使用 VIP 访问的，而各个 VM 通常是使用 VIP:<端口号>访问的。通过将 ILPIP 分配给特定 VM，可以直接使用 ILPIP 地址访问该 VM。

#### 7.1.3.1 Azure powershell 操作为 IaaS 虚拟机设置实例公共 IP 地址

需要实现的目标：

- 新建虚拟机并使用实例公共 IP 地址。
- 为已创建的虚拟机配置实例公共 IP 地址。
- 获取对应实例公共 IP 地址的域名，为自定义域名指定 CNAME 做备用。

操作步骤:

(1) 安装 Windows Azure Powershell 模块并关联 azure 订阅账户, 选择默认操作的 azure 订阅及存储账户

```
Set-AzureSubscription -SubscriptionName <订阅名称> -CurrentStorageAccountName <存储名称>
```

(2) 新建 VM 并设置实例级公共 IP 地址:

```
New-AzureVMConfig -Name <虚拟机名称> -InstanceSize Small -ImageName 55bc2b193643443bb879a78bda516fc8__Win2K8R2SP1-Datacenter-201408.01-en.us-127GB.vhd | Add-AzureProvisioningConfig -Windows -AdminUsername <虚拟机用户名> -Password <虚拟机密码> | Set-AzurePublicIP -PublicIPName <实例级公共 IP 名称> | New-AzureVM -ServiceName <云服务名称> -Location "China North"
```

(3) 通过下面的语句获取对应的 实例级公共 IP 信息 :

```
Get-AzureRole -ServiceName <云服务名称> -Slot Production -InstanceDetails | findstr "实例级公共 IP 名称"
```

### 7.1.3.2 公共实例 IP 地址相关

● 对已经存在的 VM 设置实例级公共 IP 地址:

```
Get-AzureVM -ServiceName <云服务名称> -Name <虚拟机名称> | Set-AzurePublicIP -PublicIPName <实例级公共 IP 名称> | Update-AzureVM
```

● 查询当前虚拟机是否配置了 ILPIP 地址, 需要执行以下命令:

```
PS C:\> $vm = Get-AzureVM -ServiceName 'CloudService1' -name 'VM01'
PS C:\> Get-AzurePublicIP -VM $vm
```

- 只能为每个 VM 或角色实例分配一个 ILPIP。每个订阅最多可使用 5 个 ILPIP。多 NICVM 不支持 LIPID。
- 云服务实例也可以配置 ILPIP 地址。
- 当以释放资源的方式关闭虚拟机后再启动, 实例级公共 IP 地址会发生改变, 故建议使用 DNSName 做 CNAME 的方式实现自定义域名的解析。
- ILPIP 地址适用于主动 FTP 服务器的配置, 统一出站 IP 地址的配置, 及其他需要端口全开放的服务。

参考文章:

<http://www.jonprocter.com/assign-a-public-ip-pip-to-an-azure-virtual-machine-vm/>

<https://www.azure.cn/documentation/articles/virtual-networks-instance-level-public-ip/>

### 7.1.4 静态公网 IP

针对经典环境而言, 保留 VIP 为静态公网 IP。需要使用 powershell 命令来实现。具体请参看 7.1.1 保留 IP 地址章节。

## 7.2 虚拟网络相关功能

### 7.2.1 子网划分与通信

世界互联网组织机构规定了五类 IP 地址分别为 A、B、C、D、E 五类，其中 A、B、C 是基本类，D、E 作为多播和保留使用。A 类网络有 126 个，每个 A 类网络有 16777214 台主机，它们处于同一广播域。而在同一广播域中有这么多节点是会造成很多问题的，网络会因为广播通信而饱和，结果造成 16777214 个地址大部分没有分配出去。可以把基于每类的 IP 网络进一步分成更小的网络，每个子网由路由器界定并分配一个新的子网网络地址，子网地址是借用基于每类的网络地址的主机部分创建的。划分子网后，通过使用掩码，把子网隐藏起来，使得从外部看网络没有变化，这就是子网掩码。

当我们对一个网络进行子网划分时，是为了更加方便的区分和管理网络。比如，当一组 IP 地址指定给一个公司时，公司可能将该网络拆分成更小的网络，每个部门一个。这样，技术部门和管理部门都可以有属于它们的网络。通过划分子网，我们可以按照我们的需要将网络分割成小网络。这样也有助于降低流量和隐藏网络的复杂性。

#### 7.2.1.1 子网掩码

RFC 950 定义了子网掩码的使用，子网掩码是一个 32 位的 2 进制数，其对应网络地址的所有位置都为 1，对应于主机地址的所有位置都为 0。

由此可知，A 类网络的默认子网掩码是 255.0.0.0，B 类网络的默认子网掩码是 255.255.0.0，C 类网络的默认子网掩码是 255.255.255.0。将子网掩码和 IP 地址按位进行逻辑“与”运算，得到 IP 地址的网络地址，剩下的部分就是主机地址，从而区分出任意 IP 地址中的网络地址和主机地址。

子网掩码常用点分十进制表示，我们还可以用 CIDR 的网络前缀法表示掩码，即“/<网络地址位数>”。如 138.96.0.0/16 表示 B 类网络 138.96.0.0 的子网掩码为 255.255.0.0。

#### 7.2.1.2 子网划分

在划分子网时，不仅要考虑目前需要，还应了解将来需要多少子网和主机。对子网掩码使用必须要更多的子网位，可以得到更多的子网，节约了 IP 地址资源，若将来需要更多子网时，不用再重新分配 IP 地址，但每个子网的主机数量有限；反之，子网掩码使用较少的子网位，每个子网的主机数量允许有更大的增长，但可用子网数量有限。一般来说，一个网络中的节点数太多，网络会因为广播通信而饱和，所以，网络中的主机数量的增长是有限的，也就是说，在条件允许的情况下，会将更多的主机位用于子网位。

综上所述，子网掩码的设置关系到子网的划分。子网掩码设置的不同，所得到的子网不同，每个子网能容纳的主机数目不同。若设置错误，可能导致数据传输错误。

#### 7.2.1.3 子网通信

子网掩码告知路由器，IP 地址的前多少位是网络地址，后多少位（剩余位）是主机地



址，使路由器正确判断任意 IP 地址是否是本网段的，从而正确地进行路由。

例如，有两台主机，主机一的 IP 地址为 222.21.160.6，子网掩码为 255.255.255.192，主机二的 IP 地址为 222.21.160.73，子网掩码为 255.255.255.192。现在主机一要给主机二发送数据，先要判断两个主机是否在同一网段。

主机一

222.21.160.6 即：11011110.00010101.10100000.00000110

255.255.255.192 即：11111111.11111111.11111111.11000000

按位逻辑与运算结果为：11011110.00010101.10100000.00000000

十进制形式为（网络地址）：222.21.160.0

主机二

222.21.160.73 即：11011110.00010101.10100000.01001001

255.255.255.192 即：11111111.11111111.11111111.11000000

按位逻辑与运算结果为：11011110.00010101.10100000.01000000

十进制形式为（网络地址）：222.21.160.64

## 7.2.2 用户定义路由与 IP 转发

### 7.2.2.1 用于自定义路由和 IP 转发概述

在 Azure 中将虚拟机（VM）添加到虚拟网络（VNet）时，VM 能够自动通过网络进行相互通信。你不需要指定网关，即使这些 VM 位于不同子网中。

假如客户的环境想要实现自定义的路由，而不是按照虚拟网络设计的网关来进行通信。那么我们就需要考虑用 UDR。比如过如果客户的在虚拟网络面有多台 VM，并且多台 VM 想通过另外一台 VM 或者防火墙来访问公网，从而实现流量控制和统计等功能

下图（图 7.2-1）就是 UDR 的一个应用场景：



图 7.2-1

- 从同一子网内。
- 在 VNet 中从一个子网到另一个子网。
- 从 VM 到 Internet。
- 通过 VPN 网关从一个 VNet 到另一个 VNet。
- 通过 VNet 对等互连（服务链接）从一个 VNet 到另一个 VNet。
- 通过 VPN 网关从 VNet 到本地网络。

尽管使用系统路由可以自动加快通信以便部署，但在某些情况下，需要通过虚拟设备控制数据包的路由。为此，可以通过创建用户定义的路由来指定下一个跃点，从而方便数据包流向特定的子网，并转到你的虚拟设备，同时为运行的 VM 虚拟设备启用 IP 转发。

#### 7.2.2.2 案例：

图 7.2-2 显示了用户定义的路由和 IP 转发的一个示例，它强制将数据包从一个子网发送到另一个子网，继而通过第三个子网上的虚拟设备通信。

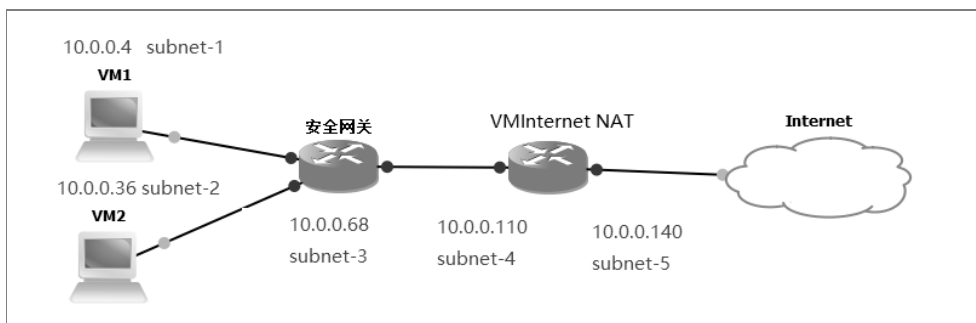


图 7.2-2

VM1 和 VM2 分别在 subnet-1 和 subnet-2，VMInternet 作为出公网的 NAT 设备，配置两个网卡一个在 subnet-4 用于接收 VM1 和 VM2 转移过来的流量，另外一个网卡在 subnet-5 用于做 NAT 并将 Internet 流量出 Azure。出公网的流量经过安全网关到 VMInternet 然后 NAT 之后出 Azure，而 VM2 到 VM1 的流量经过安全网关再到 VM1，反之亦然。

\*注意：由于先建立的网卡默认配置有 gateway 所以可以出公网，所以 NIC1 这边指定的是 10.0.0.140 这个做 NAT 的 IP。这边以 Windows Server 2008 R2 中的 RRAS 做软件 NAT。

#### 1. 通过 Powershell 创建双网卡虚拟机

```
#设置建立虚机使用的 storage account
Set-AzureSubscription -SubscriptionName 'xxxxxxx' -CurrentStorageAccountName
'xxxxxxx'
#设置虚拟机使用的 image 并指定用户名和密码
$image=get-azurevmimage | where {$_.imagename -like "*Win2K8R2SP1-
Datacenter-20160229-en.us*"}
$adminusername="xxxx"
$adminpassword="xxxx"
#设置两个 subnet
$Subnet1Name="Subnet-5"
```

```

$Subnet2Name="Subnet-4"

#设置两个网卡的 IP
$NIC1IP="10.0.0.140"
$NIC2IP="10.0.0.110"

#配置虚拟机参数
$vm = New-AzureVMConfig -Name "VMInternet" -InstanceSize "ExtraLarge" -Image
$image.ImageName
Add-AzureProvisioningConfig -VM $vm -Windows -AdminUserName $adminusername
-Password $adminpassword

#设置默认的网卡地址
Set-AzureSubnet -SubnetNames $Subnet1Name -VM $vm
Set-AzureStaticVNetIP -IPAddress $NIC1IP -VM $vm

#增加第二张网卡
Add-AzureNetworkInterfaceConfig -Name "NIC2" -SubnetName $Subnet2Name
-StaticVNetIPAddress $NIC2IP -VM $vm

#创建虚拟机
New-AzureVM -ServiceName "VMUDR" -VNetName "VNETUDR" -VM $vm -Location
'China East'

```

## 2. 配置 UDR 自定义路由走向

```

#为 subnet-1 指定路由将到 subnet-2 的流量以及到 Internet 流量指向防火墙 VM
New-AzureRouteTable -Name TestRouteTableSubnet1 -Location "China East"
-Label "Route Table Subnet1"
Get-AzureRouteTable TestRouteTableSubnet1 |Set-AzureRoute -RouteName
Route1 -AddressPrefix 10.0.0.32/27 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Get-AzureRouteTable TestRouteTableSubnet1 |Set-AzureRoute -RouteName
Route2 -AddressPrefix 0.0.0.0/0 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Set-AzureSubnetRouteTable -VirtualNetworkName VNETUDR -SubnetName "Subnet-1"
-RouteTableName TestRouteTableSubnet1

#为 subnet-2 指定路由将到 subnet-1 的流量以及到 Internet 流量指向防火墙 VM
New-AzureRouteTable -Name TestRouteTableSubnet2 -Location "China East"
-Label "Route Table Subnet2"
Get-AzureRouteTable TestRouteTableSubnet2 |Set-AzureRoute -RouteName
Route1 -AddressPrefix 10.0.0.0/27 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Get-AzureRouteTable TestRouteTableSubnet2 |Set-AzureRoute -RouteName
Route2 -AddressPrefix 0.0.0.0/0 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Set-AzureSubnetRouteTable -VirtualNetworkName VNETUDR -SubnetName "Subnet-2"
-RouteTableName TestRouteTableSubnet2

```

```
#将防火墙 VM 设置 IP forwarding 并将出到 Internet 流量指向 VMInternet
Get-AzureVM -Name VMFireWall -ServiceName VMUDR | Set-AzureIPForwarding
-Enable
New-AzureRouteTable -Name TestRouteTableVMFirewall -Location "China East"
-Label "Route Table VM Firewall"
Get-AzureRouteTable TestRouteTableVMFirewall |Set-AzureRoute -RouteName
Route1 -AddressPrefix 0.0.0.0/0 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.110
Set-AzureSubnetRouteTable -VirtualNetworkName VNETUDR -SubnetName
"Subnet-3" -RouteTableName TestRouteTableVMFirewall

#将 Internet 返回的流量指向防火墙 VM
New-AzureRouteTable -Name TestRouteTableVMInternet -Location "China East"
-Label "Route Table VM Internet"
Get-AzureRouteTable TestRouteTableVMInternet |Set-AzureRoute -RouteName
Route1 -AddressPrefix 10.0.0.0/27 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Get-AzureRouteTable TestRouteTableVMInternet |Set-AzureRoute -RouteName
Route2 -AddressPrefix 10.0.0.32/27 -NextHopType VirtualAppliance
-NextHopIpAddress 10.0.0.68
Set-AzureSubnetRouteTable -VirtualNetworkName VNETUDR -SubnetName
"Subnet-4" -RouteTableName TestRouteTableVMInternet
Set-AzureSubnetRouteTable -VirtualNetworkName VNETUDR -SubnetName
"Subnet-5" -RouteTableName TestRouteTableVMInternet
```

### 3. 配置 NAT 设备

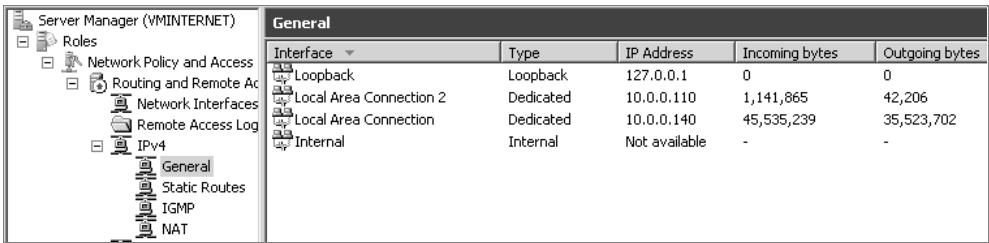
在虚机内部启用 RRAS 功能。

(1) 在 RRAS 管理界面启用 NAT 功能，如图 7.2-3 所示。



图 7.2-3

(2) 可以看到网卡的信息如图 7.2-4 所示。



Interface	Type	IP Address	Incoming bytes	Outgoing bytes
Loopback	Loopback	127.0.0.1	0	0
Local Area Connection 2	Dedicated	10.0.0.110	1,141,865	42,206
Local Area Connection	Dedicated	10.0.0.140	45,535,239	35,523,702
Internal	Internal	Not available	-	-

图 7.2-4

(3) 单击 NAT 选择 New Interface 将 10.0.0.110 这张网卡加成 private interface，请参考图 7.2-5 和图 7.2-6 所示。

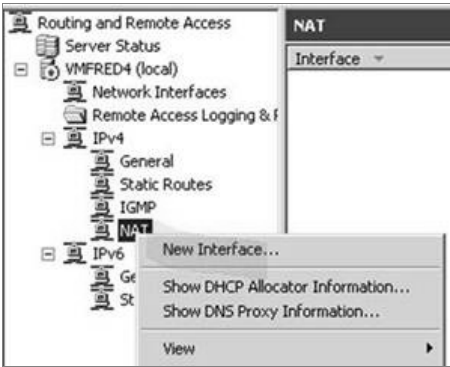


图 7.2-5



图 7.2-6

(4) 同样的，将 10.0.0.140 这张网卡添加成 Public interface 并且启用 NAT，如下图 7.2-7 所示。

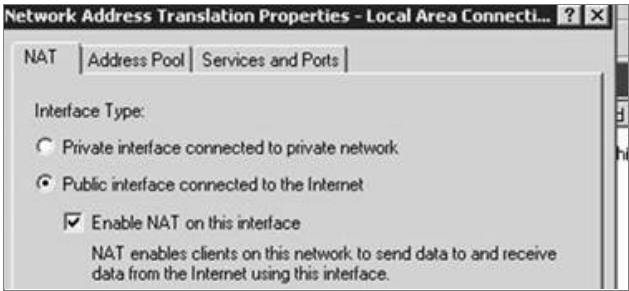


图 7.2-7

#### 4. 创建 VM1 VM2 以及安全网关所使用的虚拟机

不要将这几台虚拟机跟 VMInternet 建立在同一个云服务里面，否则会报错。

#### 5. 对安全网关所在的虚拟机内部启用 IP forwarding

Windows:修改注册表 HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\IPEnableRouter 为 1

Linux: `sysctl -w net.ipv4.ip_forward=1`

## 6. 在安全网关虚拟机内部配置一定的防火墙策略

注意：所采用的防火墙软件必须工作在网卡和 TCP/IP 层面之间，如果工作在应用层面不奏效。

## 7.3 点到站点 VPN

使用点到站点(P2S)VPN 配置可以创建从单个客户端计算机到虚拟网络的安全连接。如果要从远程位置（例如从家里或会议室）连接到 VNet，或者只有少数几个需要连接到虚拟网络的客户端，则 P2S 连接会很有用。

点到站点连接不需要 VPN 设备或面向公众的 IP 地址即可运行。可通过从客户端计算机启动连接来建立 VPN 连接。参见图 7.3-1 点到站点 VPN 架构图。

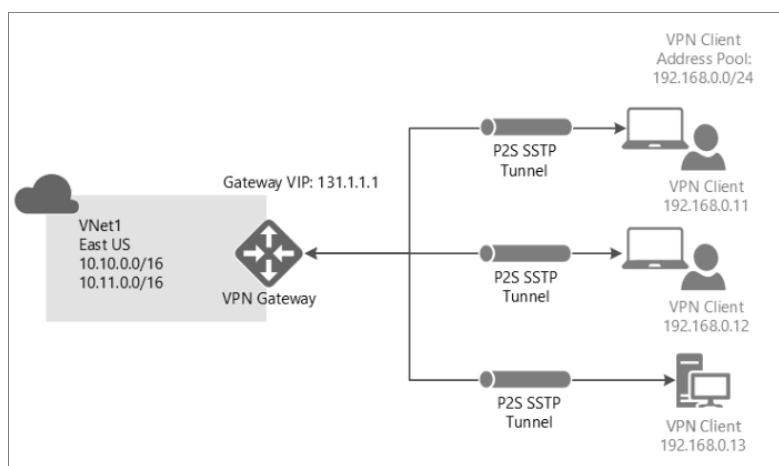


图 7.3-1

本文逐步讲解如何使用经典管理门户，在经典部署模型中创建具有点到站点连接的 VNet。点到站点连接的配置分为四个部分。必须遵循这些部分的先后顺序完成配置。请不要跳过以下步骤。

- 7.3.1 创建虚拟网络和 VPN 网关。
- 7.3.2 创建并上载用于身份验证的证书。
- 7.3.3 导出并安装客户端证书。
- 7.3.4 配置 VPN 客户端。

### 7.3.1 创建虚拟网络和 VPN 网关

#### 7.3.1.1 创建虚拟网络

(1) 登录到 Azure 经典管理门户。这些步骤使用经典管理门户而不是 Azure 门户预览。目前无法使用 Azure 门户预览创建 P2S 连接。

(2) 在屏幕左下角，单击“新建”。在导航窗格中，单击“网络服务”，然后单击“虚

拟网络”。单击“自定义创建”以启动配置向导。参见图 7.3-2 单击自定义创建。



图 7.3-2

(3) 在“虚拟网络详细信息”页上，输入以下信息，然后单击右下角的“下一步”箭头。参见图 7.3-3 虚拟网络详细信息。



图 7.3-3

- 名称—为虚拟网络命名。例如“VNet1”。将 VM 部署到此 VNet 时，需要引用此名称。
  - 位置：位置直接与用户想让资源（VM）驻留在的物理位置（区域）有关。例如，如果用户希望部署到此虚拟网络的 VM 的物理位置位于中国东部,请选择该位置。  
创建虚拟网络后，将无法更改与虚拟网络关联的区域。
- (4) 在“DNS 服务器和 VPN 连接”页上，输入以下信息，然后单击右下角的“下一步”箭头。参见图 7.3-4 DNS 服务器和 VPN 连接。

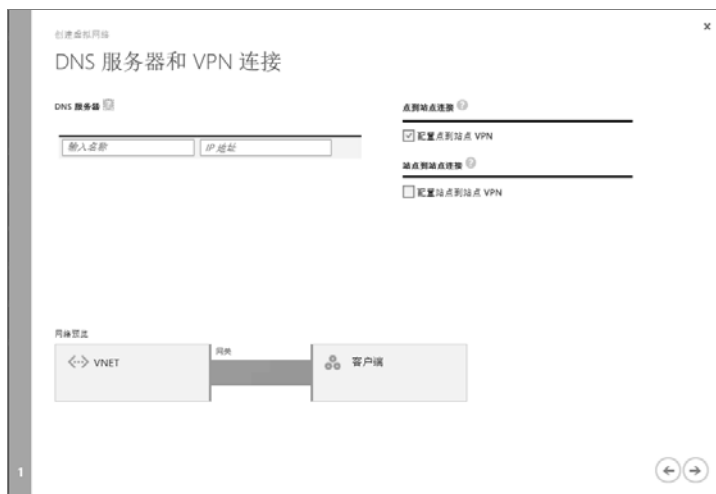


图 7.3-4

- DNS 服务器：输入 DNS 服务器名称和 IP 地址，或从快捷菜单中选择一个以前注册的 DNS 服务器。此设置不创建 DNS 服务器。此设置允许指定要用于对此虚拟网络进行名称解析的 DNS 服务器。如果用户想要使用 Azure 默认名称解析服务，请将本部分留空。
- 配置点到站点 VPN：选中此复选框。

(5) 在“点到站点连接”页上，指定用户的 VPN 客户端在连接后接收 IP 地址时的 IP 地址范围。有几个与用户能够指定的地址范围相关的规则。必须确保指定的范围与本地网络上的任何范围不重叠。参见图 7.3-5 点到站点连接。



图 7.3-5

- (6) 请输入以下信息，然后单击“下一步”箭头。
- 地址空间：包括“起始 IP”和 CIDR（地址计数）。
  - 添加地址空间：仅在网络设计需要时才添加地址空间。



(7) 在“虚拟网络地址空间”页上，指定要用于虚拟网络的地址范围。这些都是动态 IP 地址（DIPS），将分配给部署到此虚拟网络的 VM 和其他角色实例。参见图 7.3-6 虚拟网络地址空间。



图 7.3-6

所选范围不要与本地网络所用范围重叠，这一点尤其重要。必须与网络管理员协调，他们可能需要从本地网络地址空间划分一个 IP 地址范围供虚拟网络使用。

(8) 输入以下信息，然后单击复选标记即可创建虚拟网络。

- 地址空间：添加要用于此虚拟网络的内部 IP 地址范围，包括起始 IP 和计数。所选范围不要与本地网络所用范围重叠，这一点非常重要。
- 添加子网：附加的子网不是必需的，但用户可能需要为具有静态 DIP 的 VM 创建一个单独的子网。或者，用户可能需要在子网中拥有与其他角色实例分开的 VM。
- 添加网关子网：网关子网是点到站点 VPN 所必需的。单击此项可添加网关子网。网关子网仅用于虚拟网络网关。

(9) 创建虚拟网络后，可以在 Azure 经典管理门户中的“网络”页上，看到“状态”下面列出了“已创建”。创建虚拟网络后，便可以创建动态路由网关。参见图 7.3-7 网络。



图 7.3-7

7.3.1.2 创建动态路由网关

必须将网关类型配置为动态。静态路由网关无法使用此功能。

(1) 在 Azure 经典管理门户的“网络”页上，单击创建的虚拟网络，然后导航到“仪表板”页。参见图 7.3-8 虚拟网络-仪表板。



图 7.3-8

(2) 在“仪表板”页的底部，单击“创建网关”。此时会出现一条消息，询问“是否要为虚拟网络‘VNet1’创建网关”。单击“是”即可开始创建网关。创建网关最多可能需要 45 分钟。参见图 7.3-9 创建网关和图 7.3-10 成功创建网关。



图 7.3-9

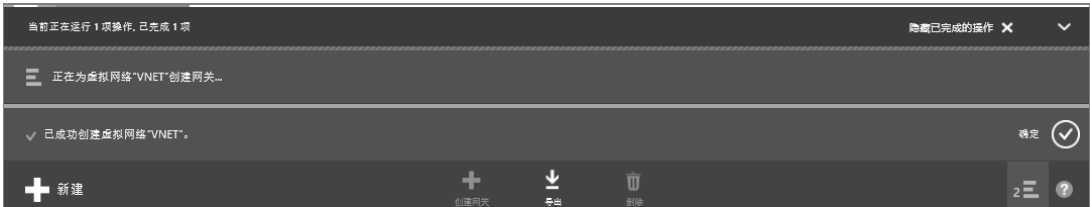


图 7.3-10

### 7.3.2 创建并上载用于身份验证的证书

证书用于对点到站点 VPN 的 VPN 客户端进行身份验证。可以使用企业证书解决方案生成的证书，或使用自签名证书。

#### 1. 下载 makecert

<https://dev.windows.com/en-us/downloads/windows-10-sdk> (supports building Windows apps and desktop applications for Windows 8.1, Windows 8, Windows 7, Windows Vista, Windows Server 2012, Windows Server 2008 R2, and Windows Server 2008.)

进入该程序目录

C:\Program Files (x86)\Windows Kits\10\bin\x64\makecert.exe

#### 2. 创建用于生成客户端证书的根证书

创建用于生成客户端证书的根证书 root certificate（用户可以将根证书的公钥和私钥导出保存以防止电脑重装系统根证书丢失）以备上传到 Azure（默认 Azure 只支持上传 20 个根证书）。参见图 7.3-11 创建根证书。

PS C:\> makecert -sky exahenge -r -n "CN=p2sroot" -pe -a sha1 -len 2048 -ss My "p2sroot.cer"

```
PS C:\> makecert -sky exahenge -r -n "CN=p2sroot" -pe -a sha1 -len 2048 -ss My "p2sroot.cer"
Succeeded
PS C:\> █
```

图 7.3-11

### 7.3.3 导出并安装客户端证书。

#### 1. 导出用于生成客户端证书的根证书

输入命令 certmgr(cmd 和 powershell 都可以，所有的用户证书均可在个人证书中找到，这是默认的存储位置）。参见图 7.3-12 输入命令 certmgr。

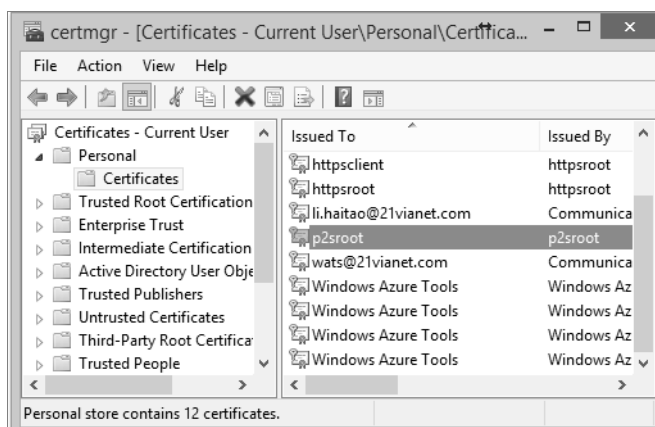


图 7.3-12

选中用户要导出的证书→右键→all tasks→export 导出私钥，参见图 7.3-13 导出私钥。

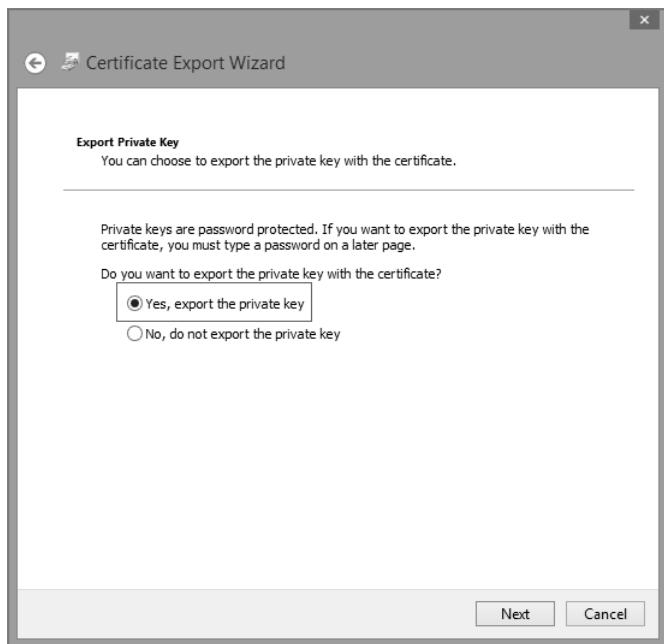


图 7.3-13

选择 yes 将私钥导出，→next-next，参见图 7.3-14 设置私钥密钥。

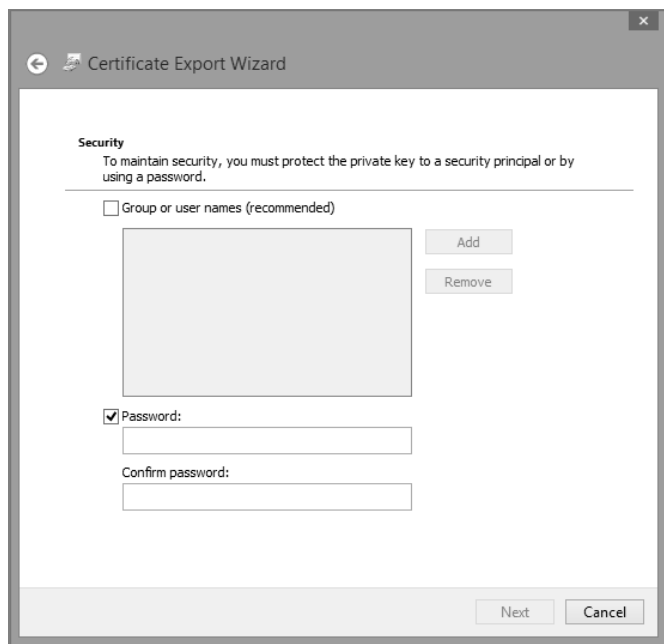


图 7.3-14

为私钥设置密钥，参见图 7.3-15 私钥导出路径。

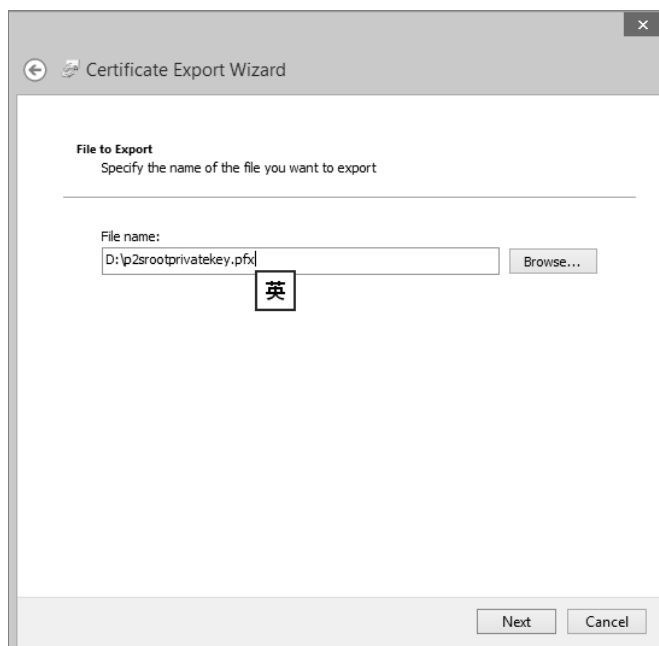


图 7.3-15

设置存储位置-next-finish（该私钥只用于保存）。

导出根证书并上传到 Azure，参见图 7.3-16 导出根证书。

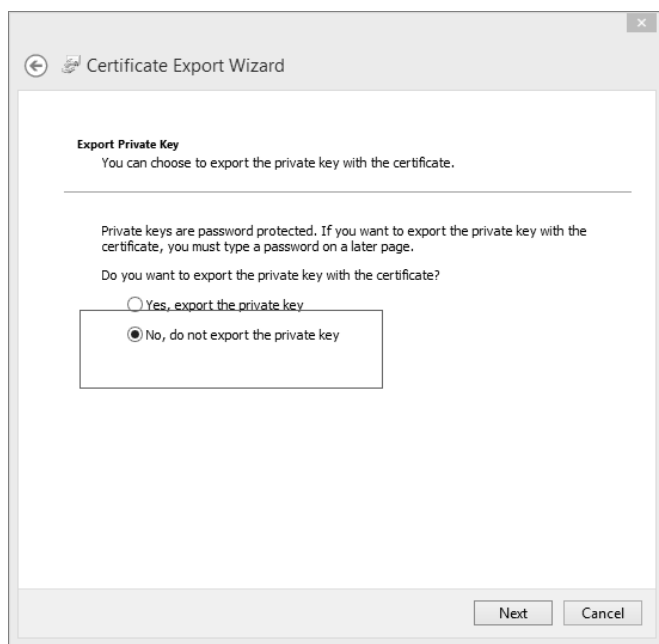


图 7.3-16

在导出 root 证书时，不导出私钥信息，其他默认即可。

## 2. 将导出的证书上传到 Azure

图 7.3-17 上传根证书和图 7.3-18 成功上传根证书。



图 7.3-17



图 7.3-18

## 3. 根据之前的根证书生成客户端证书

如果有多个客户端，建议客户端名称“p2sremoteClient”为唯一，参见图 7.3-19 生成客户端证书和图 7.3-20 成功生产客户端证书。

```
PS C:\> makecert -n "CN=p2sremoteClient" -pe -sky exchange -m 96 -ss My -in "p2sroot" -is my -a sha1
```

```
PS C:\> makecert -n "CN=p2sremoteClient" -pe -sky exchange -m 96 -ss My -in "p2sroot" -is my -a sha1
Succeeded
PS C:\>
```

图 7.3-19

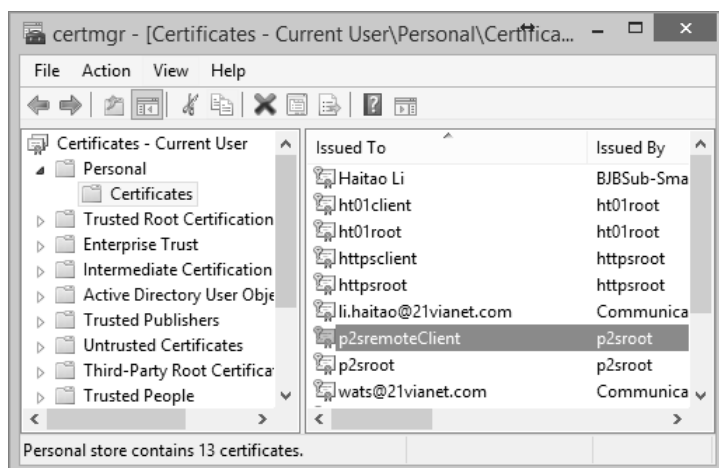


图 7.3-20

使用以上方法导出客户端的私钥，安装到需要链接 VPN 的电脑中。

#### 4. 安装客户端证书

安装方法如下：

在客户端安装导出的客户端私钥：参见图 7.3-21 双击 next 进行安装、图 7.3-22 输入私钥密码和图 7.3-23 保持默认-next-完成。

双击私钥，如图 7.3-21 所示。



图 7.3-21

Next->Next

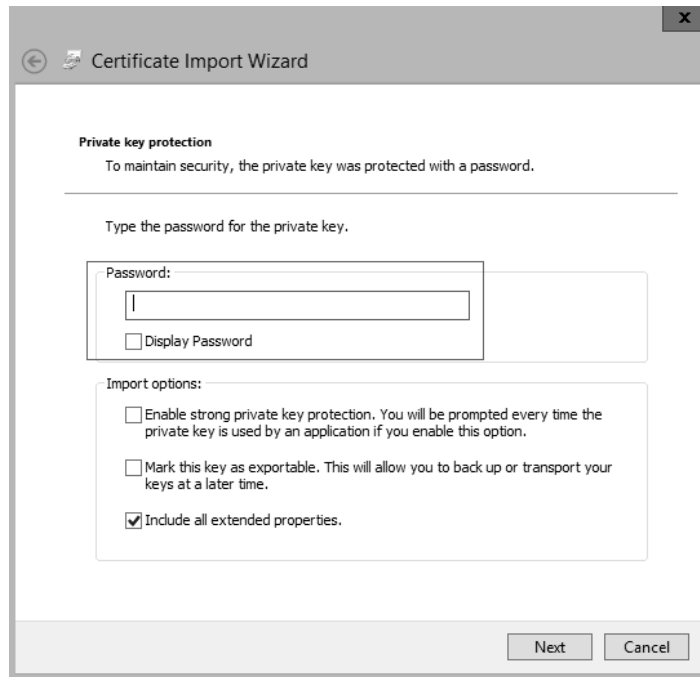


图 7.3-22

输入用户加密的私钥密码如图 7.3-23 所示。

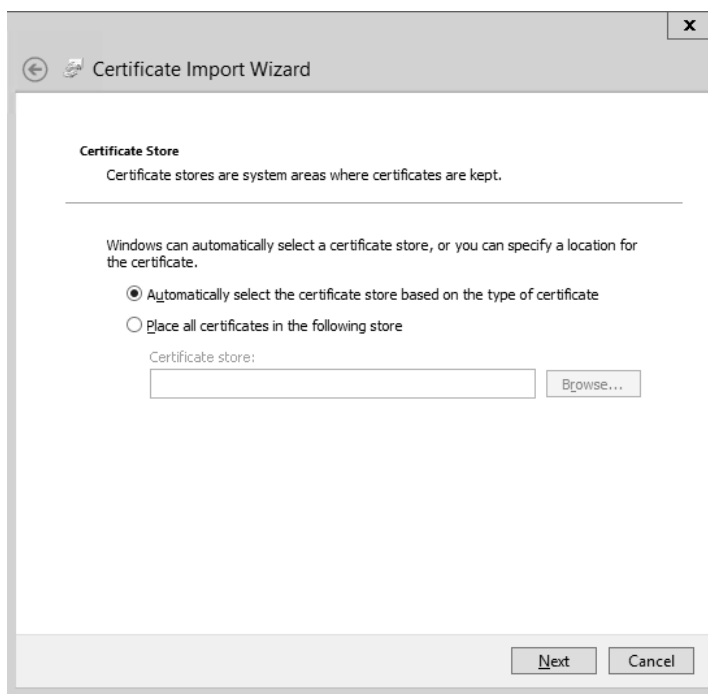


图 7.3-23



保持默认-next-完成（弹出提醒是点 yes 即可）。

5. 安装 VPN 客户端

按照自己系统的信息下载相应的客户端：  
确定系统信息方法如下（使用 cmd 和 powershell 都可以），  
通过 systeminfo 指令查看 System Type。

7.3.4 配置 VPN 客户端

1. 下载拨号器

再次单击 Azure 虚拟网络的仪表盘，在仪表盘右下角，可以看到下载 VPN 程序包，参见图 7.3-24 下载拨号器。

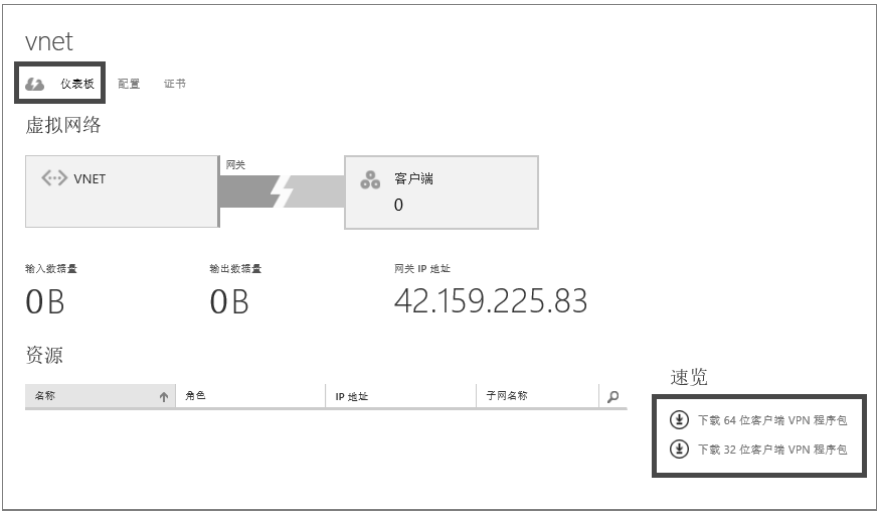


图 7.3-24

2. 安装拨号器，参见图 7.3-25 安装拨号器

- a. 双击拨号器，会弹出安装提示，选择是
- b. 安装成功以后，会在**控制面板→网络和 Internet→网络连接**中，看到已虚拟网络命名的拨号器。参见图 7.3-26 安装成功。

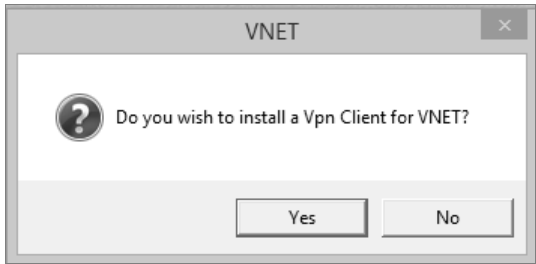


图 7.3-25

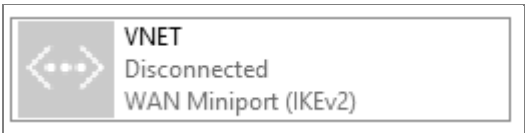


图 7.3-26

### 3. 拨号测试 P2S VPN

a. 在**控制面板**→**网络和 Internet**→**网络连接**中，双击拨号器，单击连接，参见图 7.3-27 双击拨号器。

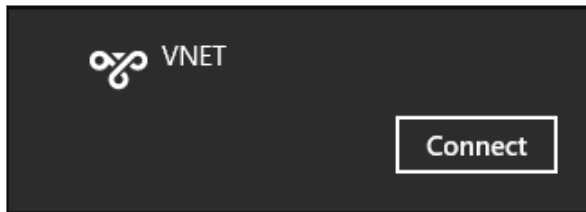


图 7.3-27

b. 在弹出的对话框中单击**连接**：参见图 7.3-28 单击连接。

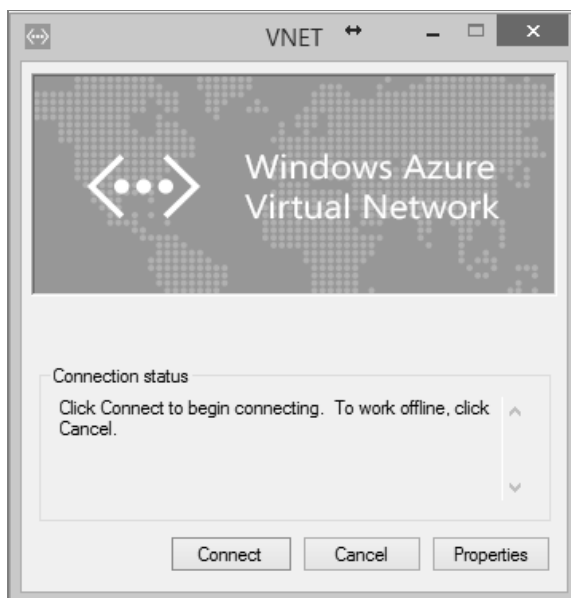


图 7.3-28

c. 连接成功后，会看到连接成功标志，参见图 7.3-29 连接成功。

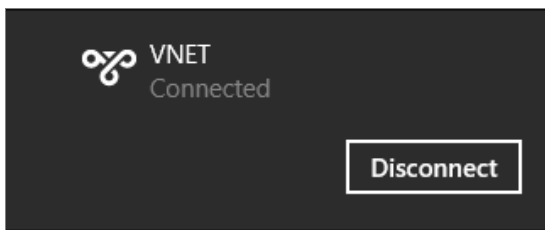


图 7.3-29

d. 同时，在 Portal 上，也能看到有一个客户端，参见图 7.3-30 虚拟网络仪表板。

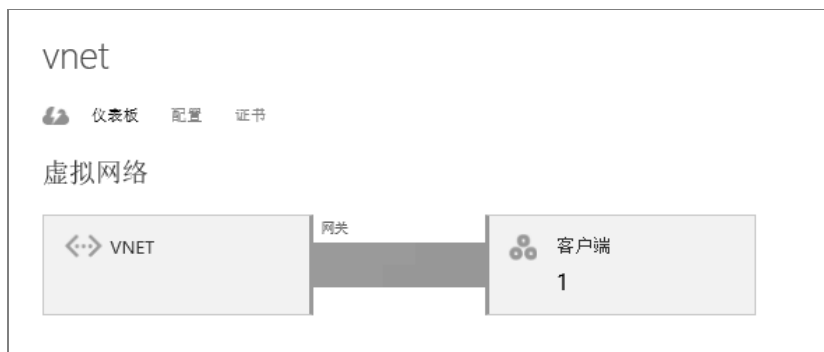


图 7.3-30

### 总结

Point-to-Site 最多可上传 20 个根证书，支持 200 个 VPN 客户端，上传到 Azure 的证书为公钥，客户端安装的证书为私钥，当用户与 VPN 交互数据时，客户的私钥加密数据，当数据到 Azure 时，Azure 使用公钥解密数据，私钥密钥公钥解密有效的标识了数字签名，数据的不可否认性，唯一性。当配置 P2S 时，Azure 网关类型为动态网关。

## 7.4 站点到站点 VPN

通过使用云端的专用网络——虚拟网络，和您本地的 on-premiss 站点，建立站点到站点的 VPN 连接，实现混合云部署。

站点到站点（S2S）VPN 网关连接是通过 IPsec/IKE（IKEv1 或 IKEv2）VPN 隧道建立连接。这种类型的连接要求 VPN 设备位于本地，并且分配有公共 IP 地址，不在 NAT 的后面。S2S 连接可用于跨界和混合配置。

需要用户注意的是，如果在 Azure 侧创建的 VNET 网关类型是静态类型（使用的是 IKEV1 协议）网关，则 Azure 侧一个 VNET 将仅支持和用户本地一个站点配置 IPsec VPN 通道，且不支持点到站点（P2S）VPN 和站点到站点（S2S）VPN 混合部署模式。如果用户在 Azure 侧创建的 VNET 网关类型为动态类型网关，用户将可以实现多个 on-premiss 网络通过 VPN 通道与一个 Azure VNET 建立多条 IPsec VPN 通道。在点到站点（P2S）VPN 和站点到站点（S2S）VPN 混合部署模式下，Azure 侧生成的网关类型默认是动态类型（使用的是 IKEV2 协议）的网关。我们将在后续章节中详细介绍虚拟网络网关的类型。

但是，目前有些 VPN 网关设备仅支持 IKEV1 协议，为了保证您的 VPN 通道可以正常建立，请在 Azure 侧生成网关时，选择生成静态类型网关。

站点到站点 VPN 网关连接的 VPN 设备，支持的参数如下：

<https://www.azure.cn/documentation/articles/VPN-gateway-about-VPN-devices/>

接下来，通过如下几个范例，读者可以了解一下通过不同类型的 VPN 设备如何将本地 on-premiss 网络和 Azure VNET 成功建立 IPsec VPN 通道、实现跨界和混合云部署。

7.4.1 使用 Windows Server 2012 搭建站点到站点 VPN

需要在 Windows Server 2012 上准备条件：两个 switch（一个 External switch、一个 Internal Switch）。

7.4.1.1 Hyper-V 创建一台虚拟机，作为 VPN 网关

创建虚拟机，暂定分配给它 1500MB 的动态内存，如图 7.4-1 所示。

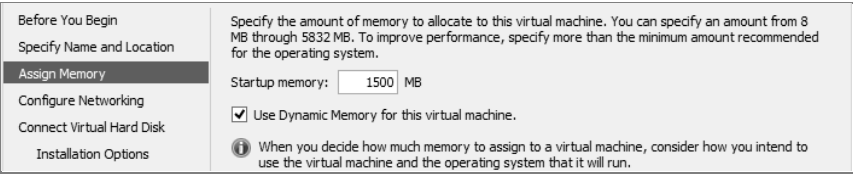


图 7.4-1

选取刚创建的 S2S VPN Switch（External Switch），如图 7.4-2 所示。

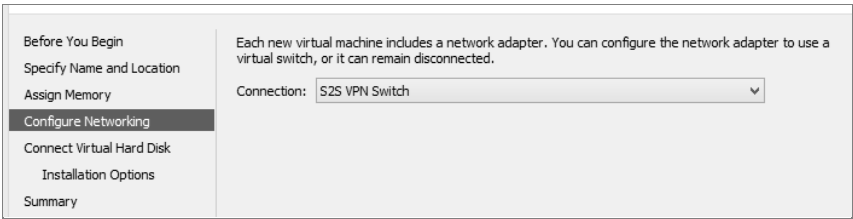


图 7.4-2

使用提前创建好的 S2SVPNmain.vhdx 虚拟硬盘，如图 7.4-3 所示。



图 7.4-3

信息汇总，选完成，如图 7.4-4 所示。

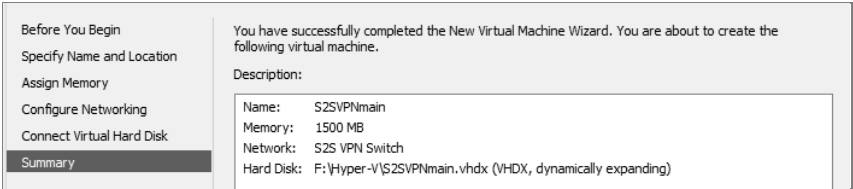


图 7.4-4

右键虚拟机，设置虚拟机，给虚拟机添加 2012 R2 的 ISO 镜像，如图 7.4-5 所示。

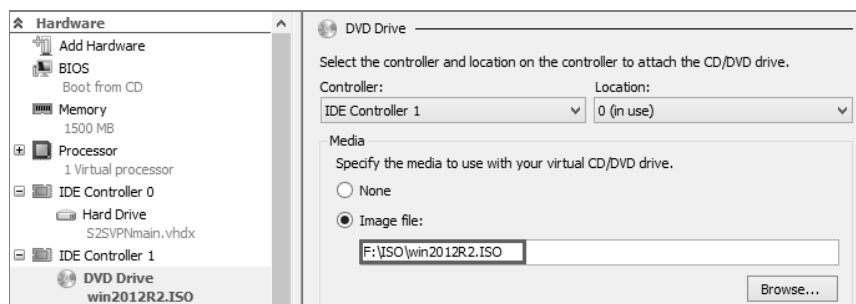


图 7.4-5

这台虚拟机需要两块网卡。

第一块网卡已经在创建虚拟机时配置了 External Switch，如图 7.4-6 所示。

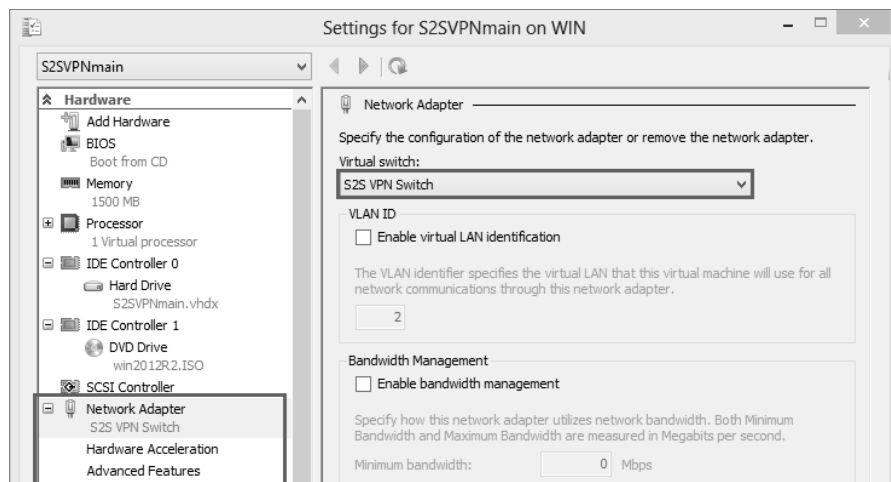


图 7.4-6

添加第二块网卡，虚拟机需要关机，因为没关机所以是灰色的，如图 7.4-7 所示。

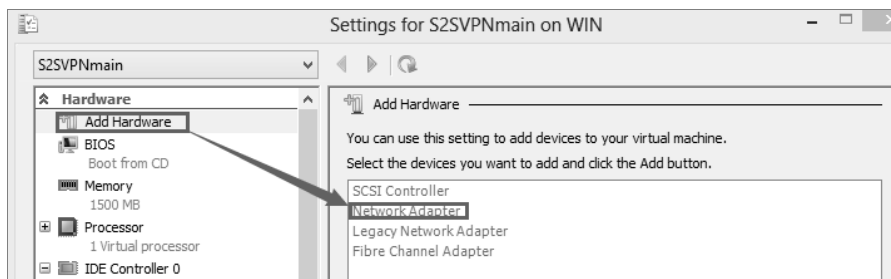


图 7.4-7

添加第二块网卡（Internal Switch），如图 7.4-8 所示。

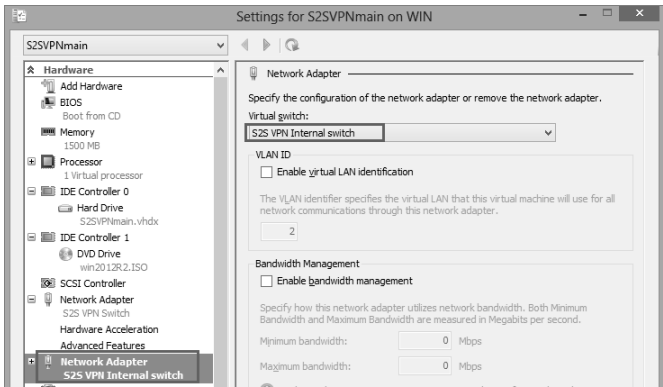


图 7.4-8

启动虚拟机，正常安装，根据需要选择 OS 版本，如图 7.4-9 所示。



图 7.4-9

测试环境，不分区，直接下一步，如图 7.4-10 所示。



图 7.4-10

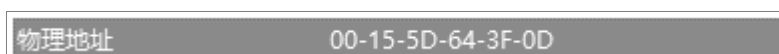
安装完成，设置密码即可，如图 7.4-11 所示。



图 7.4-11

#### 7.4.1.2 配置 VPN 网关的网卡信息

配置 S2SVPNmain 虚拟机，虚拟机内名字为“以太网”是 External Switch 下的网卡物理地址，如图 7.4-12 所示。



Hyper-v 中的地址信息



图 7.4-12

#### External 网卡配置

IP 地址为您的公网 IP 地址，默认网关请根据实际的网络拓扑酌情填写，如图 7.4-13 所示。

Internal 网卡配置，如图 7.4-14 所示。



图 7.4-13

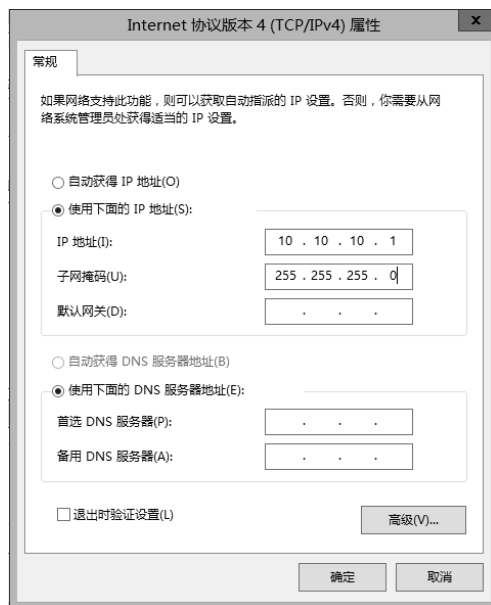


图 7.4-14

测试前，建议关闭 Windows 防火墙，否则可能 ping 不通，如图 7.4-15 所示。

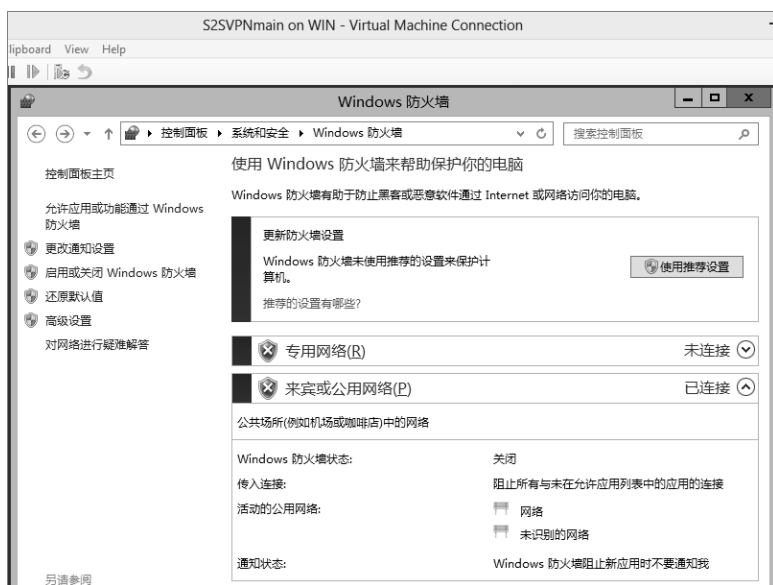


图 7.4-15

### 7.4.1.3 进行 Azure 侧的配置

第一步 创建虚拟网络，选择自定义创建，如图 7.4-16 所示。





图 7.4-16

默认下一步，选择“配置站点到站点 VPN”，如图 7.4-17 所示。



图 7.4-17

配置本地网络的地址空间，本实例中本地网络的地址空间为 10.10.10.0/24，和 VPN 设备的公网 IP 地址，红色框选处请填写 7.4.1.2 中 VPN 设备网卡的公网 IP 地址，如图 7.4-18 所示。

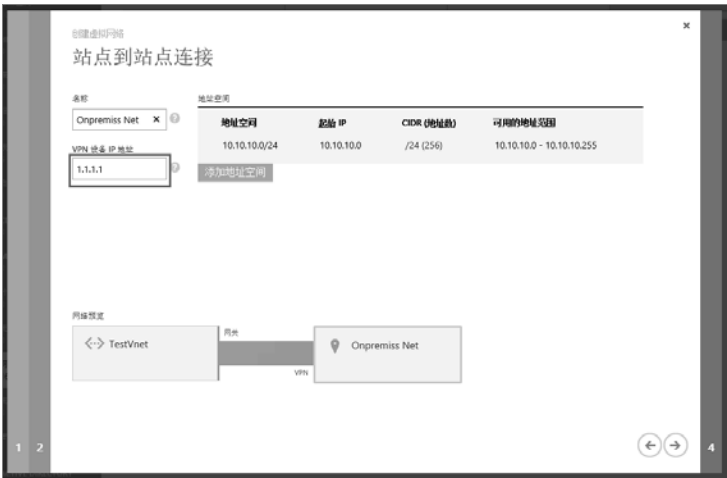


图 7.4-18

配置虚拟网络的地址空间，本实例中虚拟网络的地址空间配置为 192.0.0.0/24，如图 7.4-19 所示。



图 7.4-19

虚拟网络创建好后，请导航至刚刚创建好的虚拟网络的“仪表板”页面，在页面底部单击创建网关，创建时请选择创建动态路由网关。创建网关可能至少需要十几分钟的时间。

网关创建好后，单击“虚拟网络—仪表板”右侧的“下载 VPN 设备脚本”：请选项对应的 Windows Server 2012，如图 7.4-20 所示。



图 7.4-20

下载后配置文件为 cfg 格式的 VpnDeviceScript.cfg 文件，请改后缀为.ps1

7.4.1.4 在 Windows Server 2012 R2 运行 portal 上下载的 VPN 设备脚本

登录到第一步创建的 Windows Server 2012 R2 上，打开 Powershell。

执行 Set-ExecutionPolicy Unrestricted 可以使计算机执行脚本。

通过将上一步下载的 VpnDeviceScript.ps1 文件拖拽到 Powershell 中，运行该脚本。

脚本执行完成后，可以在路由和远程访问管理界面看到拨号已经成功，如图 7.4-21 所示。

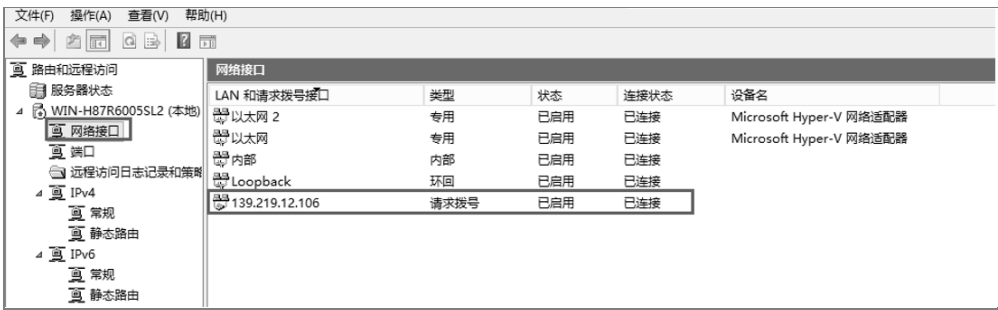


图 7.4-21

Azure 端虚拟网络仪表板显示会稍有延迟，稍后可以看到 Azure 侧显示 VPN 通道成功连接，如图 7.4-22 所示。



图 7.4-22

测试业务连通性，在 Azure 端选择一台虚拟机（IP 地址为 192.0.0.4），Onpremiss 网络内选择一台 PC 或服务器（IP 地址为 10.10.10.2），互 ping 成功。

本地长 ping 显示联通，如图 7.4-23 所示。

```
来自 192.0.0.4 的回复: 字节=32 时间=6ms TTL=126
来自 192.0.0.4 的回复: 字节=32 时间=4ms TTL=126
来自 192.0.0.4 的回复: 字节=32 时间=4ms TTL=126
来自 192.0.0.4 的回复: 字节=32 时间=6ms TTL=126
来自 192.0.0.4 的回复: 字节=32 时间=4ms TTL=126

192.0.0.4 的 Ping 统计信息:
    数据包: 已发送 = 62745, 已接收 = 62745, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 3ms, 最长 = 612ms, 平均 = 4ms
Control-C
^C
```

图 7.4-23

Azure 端长 ping 显示联通，如图 7.4-24 所示。

```

Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=6ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=7ms TTL=126
Reply from 10.10.10.2: bytes=32 time=7ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=6ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=5ms TTL=126
Reply from 10.10.10.2: bytes=32 time=6ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Reply from 10.10.10.2: bytes=32 time=4ms TTL=126
Ping statistics for 10.10.10.2:
    Packets: Sent = 63053, Received = 63052, Lost = 1 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 1839ms, Average = 4ms
Control-C
^C

```

图 7.4-24

## 7.4.2 使用 Cisco 设备建立站点到站点 VPN

本文主要介绍如何使用 Cisco 设备来建立站点到站点 VPN。关于 Azure 上面的站点到站点 VPN 的配置在这一节中就不再赘述了。

本地设备在配置 VPN 之前，首先要确定支持的 VPN 类型。

### 1. 基于策略的 VPN

PolicyBased VPN 在经典部署模型中称为静态路由网关。基于策略的 VPN 会根据使用本地网络和 Azure VNet 之间的地址前缀的各种组合配置的 IPsec 策略，加密数据包并引导其通过 IPsec 隧道。通常会在 VPN 设备配置中将策略（或流量选择器）定义为访问列表。PolicyBased VPN 类型的值为 PolicyBased。使用 PolicyBased VPN 时，请记住下列限制：

- PolicyBased VPN 仅可在基本网关 SKU 上使用。此 VPN 类型与其他网关 SKU 不兼容。
- 如果使用 PolicyBased VPN，可以只有 1 个隧道。
- 只能将 PolicyBased VPN 用于 S2S 连接且只能用于特定配置。大多数 VPN 网关配置需要 RouteBased VPN。

### 2. 基于路由的 VPN:

RouteBased VPN 以前在经典部署模型中称为动态路由网关。RouteBased VPN 使用 IP 转发或路由表中的“路由”将数据包引导到相应的隧道接口中。然后，隧道接口会加密或解密出入隧道的数据包。RouteBased VPN 的策略（或流量选择器）配置为任意到任意（或通配符）。RouteBased VPN 类型的值为 RouteBased。

我们以 Cisco ISR 2900s 为例来介绍一下如何配置这两种类型的 IPsec VPN。

- 本地虚拟网络网关 IP 地址：131.X.X.X。
- Azure 网关 IP 地址：40.76.X.X。
- 本地网络前缀：10.0.0.0/8。

- Azure 虚拟网络前缀: 192.168.1.0/16。
- Share Key: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX。

#### 7.4.2.1 基于策略的 VPN 示例

```

! Microsoft Corporation
! Windows Azure Virtual Network

! This configuration template applies to Cisco ISR 2900 Series Integrated
Services Routers running IOS 15.0.
! It configures an IPSec VPN tunnel connecting your on-premise VPN device
with the Azure gateway.

!
-----
! ACL rules
!
! Proper ACL rules are needed for permitting cross-premise network traffic.
! You should also allow inbound UDP/ESP traffic for the interface which will
be used for the IPSec tunnel.
! In this example 10.0.0.0/8 is the on premises network & 192.168.1.0/16 is
the Azure Virtual Network

access-list 101 permit ip 10.0.0.0 0.255.255.255 192.168.0.0 0.0.255.255

-----
! Internet Key Exchange (IKE) configuration
!
! This section specifies the authentication, encryption, hashing,
Diffie-Hellman, and lifetime parameters for the Phase 1 negotiation and the main
mode security association. We have picked an arbitrary policy # "10" as an
example. If that happens to conflict with an existing policy, you may choose
to use a different policy #.
! In this example the Azure Gateway IP Address is 40.76.X.X

crypto isakmp policy 10
authentication pre-share
encryption aes 256
hash sha
group 2
lifetime 28800
exit

crypto isakmp key XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX address 40.76.X.X

```

```
!
-----

! IPsec configuration
!
! This section specifies encryption, authentication, tunnel mode properties
for the Phase 2 negotiation

crypto ipsec transform-set azure-ipsec-proposal-set esp-aes 256 esp-sha-hmac
mode tunnel
exit

!
-----

! Crypto map configuration
!
! This section defines a crypto map that binds the cross-premise network
traffic to the
! IPsec transform set and remote peer. We have picked an arbitrary ID # "
10 " as an example. If that happens to conflict with an existing crypto map, you
may choose to use a different ID #.

crypto map azure-crypto-map 10 ipsec-isakmp
set peer 40.76.X.X
set security-association lifetime seconds 3600
set security-association lifetime kilobytes 102400000
set transform-set azure-ipsec-proposal-set
match address 101
exit

!
-----

! External interface configuration
!
! This section binds to the external interface of the router so that the
cross-premise network traffic matching the traffic selector defined in the crypto
map will be properly encrypted and transmitted via the IPsec VPN tunnel. It also
adjusts the TCP MSS value properly to avoid fragmentation

interface <NameOfYourOutsideInterface>
no crypto map
crypto map azure-crypto-map
ip tcp adjust-mss 1350
exit
```

注：Cisco ISR 7200 系列路由器仅支持基于策略的 VPN。

## 7.4.2.2 基于路由的 VPN 示例

```

! Microsoft Corporation
! Windows Azure Virtual Network

! This configuration template applies to Cisco ISR 2900 Series Integrated
Services Routers running IOS 15.1.
! It configures an IPSec VPN tunnel connecting your on-premise VPN device
with the Azure gateway.

! Things that begin with "azure-" are variable names and can be changed
consistently.

!
-----
! ACL rules
!
! Proper ACL rules are needed for permitting cross-premise network traffic.
! You should also allow inbound UDP/ESP traffic for the interface which will
be used for the IPSec tunnel.
! In this example 10.0.0.0/8 is the on premises network & 192.168.1.0/16 is
the Azure Virtual Network
! In this example the Azure Gateway IP Address is 40.76.X.X and your Outside
Interface IP Address is 131.X.X.X

access-list 101 permit ip 10.0.0.0 0.255.255.255 192.168.0.0 0.0.255.255
access-list 102 permit udp host 40.76.X.X eq isakmp host 131.X.X.X
access-list 102 permit esp host 40.76.X.X host 131.X.X.X

!
-----
! Internet Key Exchange (IKE) configuration
!
! This section specifies the authentication, encryption, hashing, and
Diffie-Hellman group parameters for the Phase 1 negotiation and the main mode
security association.
! In this example the Azure Gateway IP Address is 40.76.X.X

crypto ikev2 proposal azure-proposal
  encryption aes-cbc-256 aes-cbc-128 3des
  integrity sha1
  group 2

```

```
exit

crypto ikev2 policy azure-policy
proposal azure-proposal
exit

crypto ikev2 keyring azure-keyring
peer 40.76.X.X
address 40.76.X.X
pre-shared-key XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
exit
exit

crypto ikev2 profile azure-profile
match address local interface <NameOfYourOutsideInterface>
match identity remote address 40.76.X.X 255.255.255.255
authentication remote pre-share
authentication local pre-share
keyring local azure-keyring
exit

!
-----
! IPsec configuration
!
! This section specifies encryption, authentication, tunnel mode properties
for the Phase 2 negotiation

crypto ipsec transform-set azure-ipsec-proposal-set esp-aes 256 esp-sha-hmac
mode tunnel
exit

!
-----
! Crypto map configuration
!
! This section defines a crypto profile that binds the cross-premise network
traffic to the IPsec transform set and remote peer.
! We also bind the IPsec policy to the virtual tunnel interface, through which
cross-premise traffic will be transmitted.
! We have picked an arbitrary tunnel id "1" as an example. If that happens
```



to conflict with an existing virtual tunnel interface, you may choose to use a different id.

! The IP address 169.254.0.1 acts as the “inner” address of the tunnel. Essentially it has one job, to deliver traffic from the Azure side to the on-prem side. As it does not need to reach the Internet, it being routable is not necessary. The ISR has an internal routing table and knows what to do with the traffic. You should be able to use any 169.254.X.X address.

```
crypto ipsec profile azure-vti
  set transform-set azure-ipsec-proposal-set
  set ikev2-profile azure-profile
exit

int tunnel 1
  ip address 169.254.0.1 255.255.255.0
  ip tcp adjust-mss 1350
  tunnel source <NameOfYourOutsideInterface>
  tunnel mode ipsec ipv4
  tunnel destination 40.76.X.X
  tunnel protection ipsec profile azure-vti
exit

ip route 192.168.0.0 255.255.0.0 tunnel 1
```

### 7.4.3 使用非官方推荐设备建立站点到站点 VPN

如果没有看到设备在官网中的“已验证的 VPN 设备”表中列出，该设备仍有可能兼容站点到站点连接。

首先请确保本地 VPN 设备满足以下要求：

站点到站点 (S2S) VPN 网关连接是通过 IPsec/IKE (IKEv1 或 IKEv2) VPN 隧道建立的连接。这种类型的连接要求 VPN 设备位于本地，并且分配有公共 IP 地址，不在 NAT 的后面。

IPsec 参数的设定：

尽管 Azure VPN 网关支持下表中列出的值，但你目前无法从 Azure VPN 网关中指定或选择特定的组合。你必须从本地 VPN 设备指定任何约束。此外，你必须将 MSS 固定在 1350。

请根据 Azure 上 VPN 网关创建的类型，选择对应的 IPsec 参数，请参考如下链接：

<https://azure.microsoft.com/en-us/documentation/articles/VPN-gateway-about-VPN-devices/?cdn=disable>

IKE 阶段 1 设置见表 7.4-1 所示。

表 7.4-1

IKE 阶段 1 设置		
属性	PolicyBased	RouteBased 和标准或高性能 VPN 网关
SDK 版本	IKEv1	IKEv2
Diffie-Hellman 组	组 2（1024 位）	组 2（1024 位）
身份验证方法	预共享密钥	预共享密钥
加密算法	AES256 AES128 3DES	AES256 3DES
哈希算法	SHA1(SHA128)	SHA1(SHA128)、SHA2(SHA256)
阶段 1 安全关联 (SA) 生命周期（时间）	28,800 秒	10,800 秒

IKE 阶段 2 设置见表 7.4-2 所示。

表 7.4-2

IKE 阶段 2 设置		
属性	PolicyBased	RouteBased 和标准或高性能 VPN 网关
SDK 版本	IKEv1	IKEv2
哈希算法	SHA1(SHA128) SHA2(SHA256)	SHA1(SHA128)、SHA2(SHA256)
阶段 2 安全关联 (SA) 生命周期（时间）	3,600 秒	3,600 秒
阶段 2 安全关联 (SA) 生命周期（吞吐量）	102,400,000 KB	-
IPsec SA 加密和身份验证产品（按偏好顺序列出）	1.ESP-AES256 2.ESP-AES128 3.ESP-3DES 4.不适用	请参阅 RouteBased 网关 IPsec 安全关联 (SA) 产品（见下）
完全向前保密 (PFS)	否	否 (*)
对等存活检测	不支持	支持

(\*) 充当 IKE 响应方的 Azure 网关可接受 PFS DH 组 1、2、5、14、24。

**RouteBased 网关 IPsec 安全关联 (SA) 产品**

表 7.4-3 列出 IPsec SA 加密和身份验证产品。这些产品按提供或接受产品的偏好顺序列出。

表 7.4-3

RouteBased 网关 IPsec 安全关联 (SA) 产品		
IPsec SA 加密和身份验证产品	Azure 网关作为发起方	Azure 网关作为响应方
1	ESP AES_256 SHA	ESP AES_128 SHA
2	ESP AES_128 SHA	ESP 3_DES MD5
3	ESP 3_DES MD5	ESP 3_DES SHA
4	ESP 3_DES SHA	AH SHA1(具有含 null HMAC 的 ESP AES_128)
5	AH SHA1 (具有含 null HMAC 的 ESP AES_256)	AH SHA1 (具有含 null HMAC 的 ESP 3_DES)
6	AH SHA1 (具有含 null HMAC 的 ESP AES_128)	AH MD5 (具有含 null HMAC 的 ESP 3_DES)， 不建议生命周期
7	AH SHA1 (具有含 null HMAC 的 ESP 3_DES)	AH SHA1 (具有 ESP 3_DES SHA1)，无生命周期

(续表)

RouteBased 网关 IPsec 安全关联 (SA) 产品		
IPsec SA 加密和身份验证产品	Azure 网关作为发起方	Azure 网关作为响应方
8	AH MD5(具有含 null HMAC 的 ESP 3_DES), 不建议生命周期	AH MD5 (具有 ESP 3_DES MD5), 无生命周期
9	AH SHA1 (具有 ESP 3_DES SHA1), 无生命周期	ESP DES MD5
10	AH MD5(具有 ESP 3_DES MD5), 无生命周期	ESP DES SHA1, 无生命周期
11	ESP DES MD5	AH SHA1 (具有 ESP DES null HMAC), 不建议生命周期
12	ESP DES SHA1, 无生命周期	AH MD5 (具有 ESP DES null HMAC), 不建议生命周期
13	AH SHA1 (具有 ESP DES null HMAC), 不建议生命周期	AH SHA1 (具有 ESP DES SHA1), 无生命周期
14	AH MD5 (具有 ESP DES null HMAC), 不建议生命周期	AH MD5 (具有 ESP DES MD5), 无生命周期
15	AH SHA1 (具有 ESP DES SHA1), 无生命周期	ESP SHA, 无生命周期
16	AH MD5 (具有 ESP DES MD5), 无生命周期	ESP MD5, 无生命周期
17	-	AH SHA, 无生命周期
18	-	AH MD5, 无生命周期

可以对 RouteBased 和高性能 VPN 网关指定 IPsec ESP NULL 加密。基于 Null 的加密不对传输中的数据提供保护, 仅应在需要最大吞吐量和最小延迟时才使用。客户端可以在 VNet 到 VNet 通信方案中选择使用此方法, 或者在解决方案中的其他位置应用加密时使用此方法。

若要通过 Internet 建立跨界连接, 请使用默认的 Azure VPN 网关设置以及上表中列出的加密和哈希算法, 确保关键通信的安全性。

## 7.5 虚拟网络网关

### 7.5.1 动态/静态类型网关

若要在 Azure 网络和本地站点之间发送网络流量, 则必须为虚拟网络创建虚拟网络网关。VPN 网关是一种通过公共连接发送加密流量的虚拟网络网关。还可以使用 VPN 网关通过 Microsoft 网络在 Azure 虚拟网络之间发送流量。

有两种类型的虚拟网络网关: “ExpressRoute” 和 “VPN”。创建虚拟网络网关时, 需指定要创建的网关类型。VPN 网关是使用 “VPN” 网关类型的虚拟网络网关。

每个虚拟网络可以拥有两个虚拟网络网关, 但只能有一种类型。根据所选设置, 可以创建到单个 VPN 网关的多个连接。其示例之一是多站点连接配置。创建同一 VPN 网关的

多个连接时，所有 VPN 隧道（包括点到站点 VPN）共享网关可用的带宽。

#### 7.5.1.1 先介绍概念

##### VPN 类型

为 VPN 网关配置创建虚拟网络网关时，必须指定 VPN 类型。选择的 VPN 类型取决于要创建的连接拓扑。例如，P2S 连接需要 RouteBased VPN 类型。VPN 类型还取决于要使用的硬件。S2S 配置需要 VPN 设备。有些 VPN 设备仅支持特定的 VPN。

选择的 VPN 类型必须满足所要创建的解决方案的所有连接要求。例如，如果要为同一虚拟网络创建 S2S VPN 网关连接和 P2S VPN 网关连接，应使用 VPN 类型 RouteBased，因为 P2S 需要 RouteBased VPN 类型。此外，需确认 VPN 设备支持 RouteBased VPN 连接。

创建虚拟网络网关后，无法更改 VPN 类型。必须删除虚拟网络网关，然后新建一个，有以下两种 VPN 类型。

(1) PolicyBased: PolicyBased VPN 在经典部署模型中称为静态路由网关。基于策略的 VPN 会根据使用本地网络和 Azure VNet 之间的地址前缀的各种组合配置的 IPsec 策略，加密数据包并引导其通过 IPsec 隧道。通常会在 VPN 设备配置中将策略（或流量选择器）定义为访问列表。PolicyBased VPN 的值为 PolicyBased。使用 PolicyBased VPN 时，请记住下列限制：

- PolicyBased VPN 仅可在基本网关 SKU 上使用。此 VPN 类型与其他网关 SKU 不兼容。
- 如果使用 PolicyBased VPN，可以只有 1 个隧道。
- 只能将 PolicyBased VPN 用于 S2S 连接且只能用于特定配置。大多数 VPN 网关配置需要 RouteBased VPN。

(2) RouteBased: RouteBased VPN 以前在经典部署模型中称为动态路由网关。RouteBased VPN 使用 IP 转发或路由表中的“路由”将数据包引导到相应的隧道接口中。然后，隧道接口会加密或解密出入隧道的数据包。RouteBased VPN 的策略（或流量选择器）配置为任意到任意（或通配符）。RouteBased VPN 类型的值为 RouteBased。

#### 7.5.1.2 配置经典部署模型的 VPN 网关

如果想要在 Azure 与本地位置之间创建安全的跨界连接，则需要配置 VPN 网关连接。在经典部署模型中，网关可以是以下两种 VPN 路由类型之一：静态或动态。所选的类型取决于用户的网络设计规划和想要使用的本地 VPN 设备。

例如，某些连接选项（如点到站点连接）需要动态路由网关。如果想要将网关配置为同时支持点到站点（P2S）连接和站点到站点（S2S）连接，即使可使用任意一种网关 VPN 路由类型配置站点到站点，也必须要配置动态路由网关。

以下步骤将演示如何在 Azure 经典管理门户中配置 VPN 网关。这些步骤适用于使用经典部署模型创建的虚拟网络的网关。

##### 开始之前

在配置网关之前，必须先创建虚拟网络。有关为跨界连接创建虚拟网络的步骤，请参

阅使用站点到站点 VPN 连接配置虚拟网络或使用点到站点 VPN 连接配置虚拟网络。然后，按照以下步骤来配置 VPN 网关，并收集配置 VPN 设备所需的信息。

创建 VPN 网关

- (1) 在 Azure 经典管理门户中的“网络”页上，验证虚拟网络的状态列是否为“已创建”。
- (2) 在“名称”列中，单击用户的虚拟网络的名称。
- (3) 在“仪表板”页上，请注意此 Vnet 尚未配置网关。当用户完成配置网关的步骤时，用户将会看到此状态。如图 7.5-1 所示。

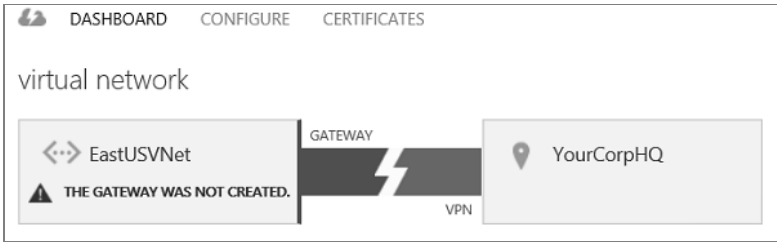


图 7.5-1

接下来，在页面底部单击“创建网关”。可以选择“静态路由”或“动态路由”。选择的 VPN 路由类型取决于几个因素。例如，VPN 设备支持的类型，以及是否需要支持点到站点连接。创建网关后，必须先删除并重新创建网关才能更改网关 VPN 路由类型。系统提示用户确认要创建网关时，单击“是”。如图 7.5-2 所示。



图 7.5-2

正在创建网关时，请注意页面上的网关图形将更改为黄色，并显示“正在创建网关”。创建网关最多可能需要 45 分钟。等到网关创建完成才能继续进行其他配置设置。如图 7.5-3 所示。

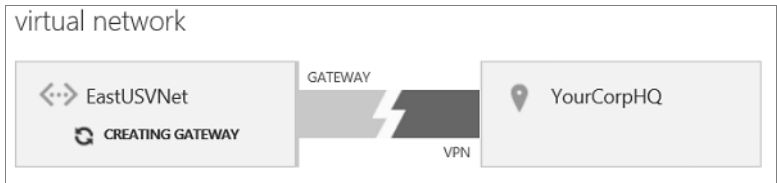


图 7.5-3

当网关状态更改为“正在连接”时，用户可以收集 VPN 设备所需的信息。如图 7.5-4 所示。

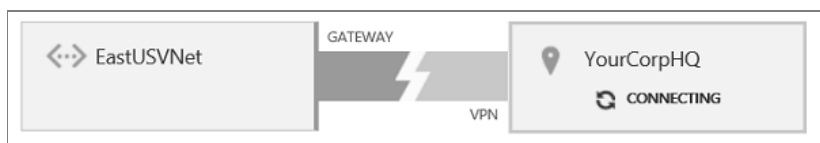


图 7.5-4

#### 收集 VPN 设备配置信息

创建网关后，收集 VPN 设备配置的信息。此信息位于虚拟网络的“仪表板”页：

网关 IP 地址- IP 地址可在“仪表板”页中找到。只有在网关创建完成之后才能看到该地址。

共享密钥 - 单击屏幕底部的“管理密钥”。单击密钥旁边的图标将密钥复制到剪贴板，然后粘贴并保存密钥。仅当只有一个 S2S VPN 隧道时，此按钮才有效。如果有多个 S2S VPN 隧道，请使用获取虚拟网络网关共享密钥 API 或 PowerShell cmdlet，如图 7.5-5 所示。

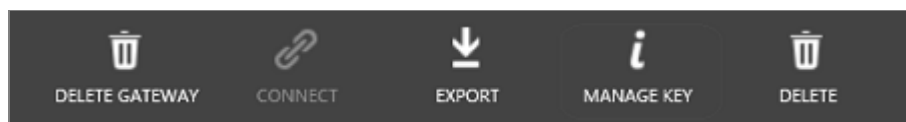


图 7.5-5

#### 7.5.1.3 如何更改网关的 VPN 路由类型

由于某些连接配置仅适用于特定网关路由类型，用户可能需要更改现有 VPN 网关的 VPN 路由网关类型。例如，用户可能要将点到站点连接添加到具有静态网关的现有站点到站点连接。点到站点连接需要动态网关。这意味着若要配置 P2S 连接，必须将网关 VPN 路由类型从静态更改为动态。

如果需要更改网关 VPN 路由类型，需要删除现有网关，然后使用新的路由类型重新创建网关。不需要删除整个虚拟网络来更改网关路由类型。

在更改网关 VPN 类型之前，请确保证验证 VPN 设备可支持所要使用的路由类型。

#### IMPORTANT:

删除虚拟网络 VPN 网关时，将释放分配给该网关的 VIP。重新创建网关时，系统会为它分配新的 VIP。

#### 7.5.1.4 删除现有 VPN

在虚拟网络的“仪表板”页上，导航到页面底部，然后单击“删除网关”。等待出现已完成删除网关的通知。当屏幕上出现已删除网关的通知后，便可以创建新网关。

### 7.5.2 高性能网关

为了给跨云和内部部署的连接提供更高的吞吐量和更多的 S2S VPN 隧道，微软发布

了新的 Azure 虚拟网络网关 SKU：高性能网关。表 7.5-1 所示为当前网关和高性能网关的初始吞吐量合计值和 S2S VPN 隧道规格：

表 7.5-1

网关 SKU	ExpressRoute 吞吐量*	S2S VPN 吞吐量*	最大 S2S 隧道数
默认	~500Mbps	~80Mbps	10 个
高性能	~1000Mbps	~200Mbps	30 个

\* 请注意，实际吞吐量会因流量状况和应用程序行为的不同而发生变化。

### 7.5.2.1 先介绍下概念

#### 网关 SKU

创建虚拟网络网关时，需要指定要使用的网关 SKU。如果选择更高级的网关 SKU，则将为该网关分配更多的 CPU 和网络带宽，这样使网关能够支持到虚拟网络更高的吞吐量。

VPN 网关可以使用以下 SKU：

基本

标准

HighPerformance

选择 SKU 时，请考虑以下内容：

如果想要使用 PolicyBased VPN 类型，必须使用基本 SKU。任何其他 SKU 均不支持 PolicyBased VPN（之前称为静态路由）；

基本 SKU 不支持 BGP；

基本 SKU 不支持 ExpressRoute-VPN 网关共存配置；

主动-主动 S2S VPN 网关连接只能在 HighPerformance SKU 上配置。

### 7.5.2.2 配置网关 SKU

在 Azure 门户预览中指定网关 SKU

如果使用 Azure 门户预览创建 Resource Manager 虚拟网络网关，可以使用下拉列表选择网关 SKU。显示的选项对应于所选的网关类型和 VPN 类型。

例如，如果选择“VPN”作为网关类型，选择“基于策略”作为 VPN 类型，则只会看到“基本”SKU，因为这是 PolicyBased SKU 唯一可用的 SKU。如果选择“基于路由”，则可以从“基本”、“标准”和“高性能”SKU 中选择。

使用 PowerShell 指定网关 SKU。

以下 PowerShell 示例将-GatewaySku 指定为 *Standard*。

高性能网关同时可用于 Azure 动态路由网关和 Azure ExpressRoute。不支持静态路由网关。以下 cmdlet 可用于创建新的高性能网关或者将现有网关升级到新 SKU：

### 7.5.2.3 创建高性能网关

Azure PowerShell cmdlet 中增加了一个新的选项 New-AzureVNetGateway，用于指定 SKU。以下示例将为虚拟网络“MyAzureVNet”创建高性能网关：

```
PS D:\> New-AzureVNetGateway -VNetNameMyAzureVNet -GatewayTypeDynamicRouting -GatewaySKUHighPerformance
```

请注意，DynamicRouting 同时是 DynamicRouting 网关和专用（ExpressRoute）网关的 GatewayType。因此，该示例 cmdlet 也可用于创建虚拟网络网关，以连接 ExpressRoute 线路。

#### 7.5.2.4 更新网关 SKU

以下 Resize-AzureVNetGateway cmdlet 可以更新 Azure 虚拟网络网关的 SKU：

```
PS D:\> Resize-AzureVNetGateway -VNetNameMyAzureVNet -GatewaySKUHighPerformance
```

该示例 cmdlet 会将 MyAzureVNet 的网关从 Default 更改为 HighPerformance。用户也可以将网关 SKU 从 High Performance 改回到 Default：

```
PS D:\> Resize-AzureVNetGateway -VNetNameMyAzureVNet -GatewaySKUDefault
```

## 7.6 Express Route

### 7.6.1 Azure 提供的 Express Route 服务简介

除了通过公网传输数据的 VPN，Azure 还提供更高安全性、更低延迟的专线接入（Express Route）访问。在 Azure 侧开通 Express Route 服务之前，首先需要向电信申请开通 Express Route 专线，连接 On-Premiss 网络到 Azure 中国东部或中国北部数据中心。On-premiss 网络前端需要有 1 台或 2 台支持 BGP 协议的设备，用以连接电信 Express Route 的主/备线路。

开通并使用 Express Route 需要如下几个主要操作步骤：

（1）向电信申请开通 Express Route，为了保障您的 SLA，请向电信申请开通两条（即一主一备）线路。

（2）在 Azure 侧创建 Express Route Circuit，并将生成的 Service Key 以及您配置 VLAN 号给电信，请电信将线路的状态置为 Provisioned 状态，操作步骤参见如下链接：

第一步：创建线路：

<http://www.windowsazure.cn/documentation/articles/expressroute-howto-circuit-classic>

第二步：配置路由：<http://www.windowsazure.cn/documentation/articles/expressroute-howto-routing-classic>

（3）配置 On-Premiss BGP 设备，有关路由器的配置请参考如下链接：<https://azure.microsoft.com/en-us/documentation/articles/expressroute-config-samples-routing/>

（4）On-Premiss BGP 设备配置好，底层线路调通后，请将虚拟网络链接到 Express Route 线路，以实现 Azure 虚拟网络下的虚拟机和 On-Premiss 站点之间的通信：

绑定 VNET，请参考如下链接：

<http://www.windowsazure.cn/documentation/articles/expressroute-howto-linkvnet-classic>



### 7.6.2 Azure 提供的 Express Route 类型

Express Route 分为标准服务和带有高级版附加组件的服务。

两种类型服务的区别如下：

(1) 带有高级版附加组件的服务，公共对等项和私有对等项的路由表限制由 4000 个路由增加为 10 000 个路由。

(2) 带有高级版附加组件的服务，增加了可连接到 ExpressRoute 线路的 VNet 的数量（默认值为 10），参见表 7.6-1。

表 7.6-1

线路带宽	Standard 版服务支持的 VNET 链接数	Premium 版服务支持的 VNET 链接数
50Mbps	10	20
100Mbps	10	25
200Mbps	10	25
500Mbps	10	40
1Gbps	10	50
2Gbps	10	60
5Gbps	10	75
10Gbps	10	100

(3) 普通版 Express Route，只能将 VNET 链接到 Azure 对应数据中心的 Express Route 通道；而带有高级版附加组件的 Express Route 服务，则可以跨 Azure 数据中心链接 VNet。示例：比如可以将在 Azure 中国北部创建的 VNet 链接到在 Azure 中国东部创建的 ExpressRoute 线路上。需要注意的是，将中国北部的 ER 通道链接到中国东部的 ER 线路后，中国北的 VNET 可以和中国东的 VNET 互相访问。

### 7.6.3 Azure 提供的 Express Route 关键参数说明：

#### Peer ASN:

Microsoft Azure 使用 AS 12076 用于 Azure 公共和 Azure 专用对等互联。Azure 保留了 AS 65515-65520 供内部使用。支持 16 和 32 位 AS 编号。用户可以使用专用 AS 编号建立到 Azure 的专用对等互连。建议用户使用 65001~65534 范围的 AS 编号进行配置。注意，65515-65520 为 Azure 保留的 ASN 编号，请务必不要使用 65515-65520 及 12076。

#### Primary subnet & Secondary subnet:

Azure 专用对等互连的 IP 地址：

你可以使用专用 IP 地址或公共 IP 地址来配置对等互连。用于配置路由的地址范围不得与用于在 Azure 中创建的虚拟网络或 On-Premiss 网络的地址范围重叠。

- 必须为路由接口保留一个/29 子网或两个/30 子网。
- 用于路由的子网可以是专用 IP 地址或公共 IP 地址。
- 子网不得与客户保留用于 Microsoft 云的范围冲突。
- 如果使用/29 子网，它将拆分成两个/30 子网。

- 第一个/30 子网用于主链路，第二个/30 子网用于辅助链路。
- 对于每个/30 子网，On-Premiss 必须在路由器上使用/30 子网的第一个 IP 地址。Azure 使用/30 子网的第二个 IP 地址来设置 BGP 会话。
- 只有设置了两个 BGP 会话，Azure Express Route 可用性 SLA 才有效。

专用对等互连示例：

如果用户选择使用 a.b.c.d/29 来设置对等互连，它将拆分成两个 /30 子网。示例如下：

- 192.168.100.128/30 将分配给 link1（提供商使用 192.168.100.129，而 Azure 使用 192.168.100.130）。
- 192.168.100.132/30 将分配给 link2（提供商使用 192.168.100.133，而 Azure 使用 192.168.100.134）。

#### **VLAN ID:**

VLAN ID 是客户自定义的内部 VLAN (C-tag) 号。不同的客户可以使用相同的 VLAN ID (C-tag)，对于运营商来说，会有一个唯一的 S-tag 进行标记，不需要用户配置网络底层的 S-tag。用户可以使用 1 至 4094 之间的数字配置 VLAN ID：

#### **Shared key:**

共享密钥，也可以不为 ER 线路配置共享密钥。

#### **Azure Private（专用对等互连）：**

可以通过专用对等域来连接虚拟网络内部署的 Azure 计算服务（即虚拟机（IaaS）和云服务（PaaS））。专用对等域被视为进入 Azure 的核心网络的受信任扩展。可以在核心网络和 Azure 虚拟网络（VNet）之间设置双向连接。这样，用户便可以使用 On-Premiss 专用 IP 地址直接连接到虚拟机和云服务。可以将多个虚拟网络连接到专用对等域。

#### **Azure Public（公共对等互连）：**

Azure 存储空间、SQL 数据库和 Web 应用等服务是使用公共 IP 地址提供的。启用公共对等互连后，用户将能够通过 Express Route 线路连接到所有 Azure 服务，而无需通过 Internet 连接。

需要用户注意的是，Azure 的 Express Route 自 2017 年 3 月 1 日起，将仅允许通过资源管理器（ARM）模式进行配置，已经在经典模式下开通的 Express Route 服务不受影响。有关 ARM 模式下如何配置 Express Route，请您参见后续章节。但是用户依然可以将经典模式（ASM）下的虚拟网络链接到 Express Route。将经典模式（ASM）的虚拟网络链接到 Express Route 需要使用如下 Azure Powershell ARM 模式命令 Enable 兼容经典模式功能：

```
$circuit = Get-AzureRMExpressRouteCircuit -Name "TestER" -ResourceGroupName "TestER"
$circuit.AllowClassicOperations = $true
Set-AzureRMExpressRouteCircuit -ExpressRouteCircuit $circuit
```

针对已经在经典模式下开通的 Express Route 线路，用户还可以使用如下 Azure Powershell 命令将其迁移至 ARM 模式：

```
#ARM 模式下登录订阅：
Login-AzureRmAccount -EnvironmentName AzureChinaCloud
#选择默认订阅：
```

```

Select-AzureSubscription -SubscriptionName "xxxx"
#创建 ARM 资源组:
New-AzureRmResourceGroup -Name "TestER" -Location "China North"
#迁移 Express Route 到 ARM 模式下:
Move-AzureRmExpressRouteCircuit -Name "TestER" -ResourceGroupName "TestER"
-location "China North" -ServiceKey xxxxxxxxxxxxxxxxxxxxxx

```

## 7.7 Express Route 混合 VPN

Express Route 混合 VPN 支持如下两种形式的网络拓扑:

### 线路冗余型:

将站点到站点 VPN 配置为 Express Route 的故障转移路径:

用户可以将站点到站点 VPN 连接配置为 Express Route 的备份。这仅适用于链接到 Azure 专用对等路径的虚拟网络。对于可通过 Azure 公共线路访问的服务，没有基于 VPN 的故障转移解决方案。Express Route 线路始终是主链接。仅当 ExpressRoute 线路失败时，数据才会流经站点到站点 VPN 路径。请参考图 7.7-1。

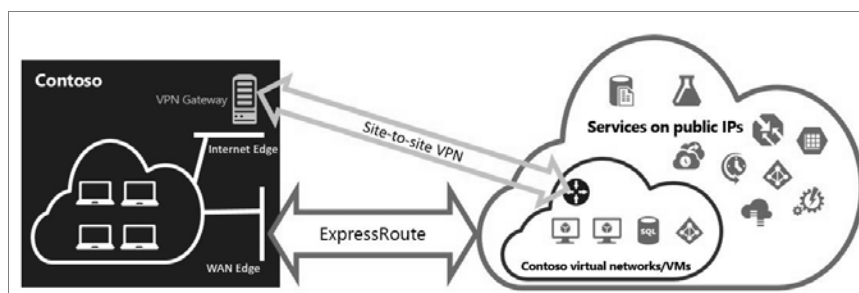


图 7.7-1

通过 VPN 连接其它 On-Premises 站点:

用户可以配置为使得部分站点通过站点到站点 VPN 直接连接到 Azure，部分站点通过 ExpressRoute 进行连接到 Azure。请参考图 7.7-2。

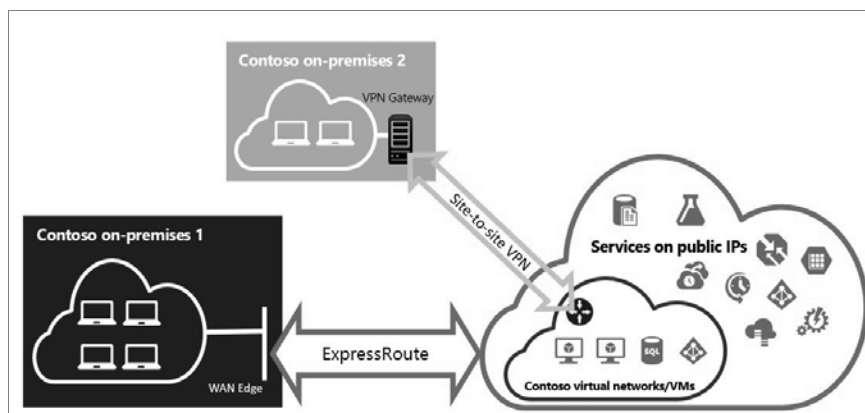


图 7.7-2

关于 Express Route 混合 VPN，需要注意如下三点：

1. Express Route 混合 VPN 配置中，不能将 VNET 配置为中转路由。
2. 使用 Express Route 混合 VPN 配置，请提前规划虚拟网络，将虚拟网络的网关子网配置为/28（或更多地址空间/27），而不要使用/29 位网关子网地址空间。
3. 配置 Express Route 混合 VPN 配置时，如果已经存在 VPN，需要先将以存在的 VPN 删除后再创建 Express Route 网关，然后再创建 VPN 网关。

更多详情，请参见如下链接：

<https://www.azure.cn/documentation/articles/expressroute-howto-coexist-classic/>

## 7.8 BGP VPN 介绍和使用

### 7.8.1 BGP 概述

BGP 是通常在 Internet 上使用的，用于在两个或更多网络之间交换路由和可访问性信息的标准路由协议。BGP 允许 Azure VPN 网关和本地 VPN 设备（称为 BGP 对等节点或邻居）交换“路由”，这些路由将通知这两个网关这些前缀的可用性和可访问性，以便这些前缀可通过涉及的网关或路由器。BGP 还可以通过将 BGP 网关从一个 BGP 对等节点获取的路由传播到所有其他 BGP 对等节点来允许在多个网络之间传输路由。BGP 是可用于 Azure 基于路由的 VPN 网关的可选功能。在启用此功能之前，你还应确保本地 VPN 设备支持 BGP。你可以继续使用不带 BGP 的 Azure VPN 网关和本地 VPN 设备。它等效于在你的网络与 Azure 之间使用静态路由（不带 BGP）与使用带 BGP 的动态路由。

BGP 的优点：

支持自动和灵活的前缀更新：

使用 BGP，你只需通过 IPsec S2S VPN 隧道为特定 BGP 对等节点声明最小前缀。它最小可为本地 VPN 设备的 BGP 对等节点 IP 地址的主机前缀（/32）。你可以控制要将哪些本地网络前缀播发到 Azure 以允许 Azure 虚拟网络访问。

你还可以播发更大的前缀，可以包括一些 VNet 地址前缀，如大型专用 IP 地址空间（例如，10.0.0.0/8）。但请注意，这些前缀不能与任一 VNet 前缀相同。与 VNet 前缀相同的这些路由将被拒绝。

支持 VNet 与本地站点之间的多个隧道基于 BGP 自动进行故障转移：

你可以在同一位置的 Azure VNet 和本地 VPN 设备之间建立多个连接。在主-主配置中，此功能在两个网络之间提供多个隧道（路径）。如果其中一个隧道断开连接，则将通过 BGP 撤消相应的路由，流量将会自动转移到其余隧道。

图 7.8-1 显示了此高可用设置的简单示例。

支持本地网络与多个 Azure VNet 之间的传输路由：

BGP 使多个网关可以从不同网络获知和传播前缀，而无论它们是直接还是间接连接。这可以为本地站点之间或跨多个 Azure 虚拟网络的 Azure VPN 网关启用传输路由。

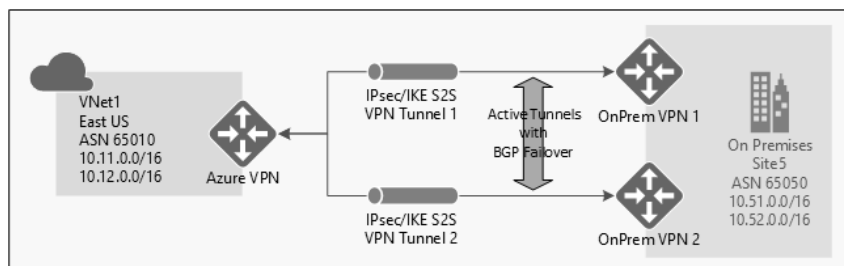


图 7.8-1

图 7.8-2 显示了多跃点拓扑的示例,其中的多个路径可以通过 Microsoft 网络中的 Azure VPN 网关在两个本地网络之间传输流量:

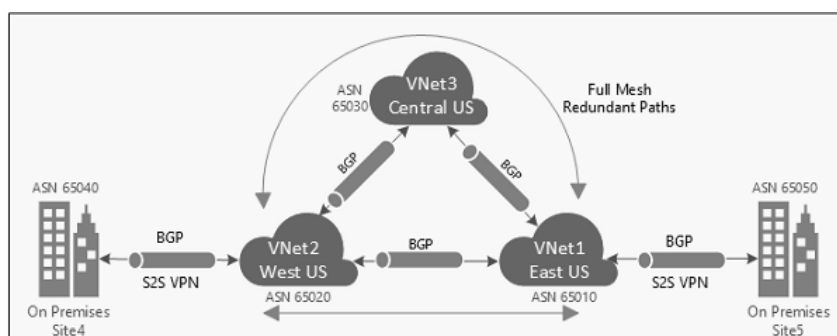


图 7.8-2

BGP 在 Azure 中的应用与下图 7.8-3 中 Cisco Secure eBGP Session with an IPsec VTI 类似,只是在 Azure 端目前无法将数据流量通过公网传输。

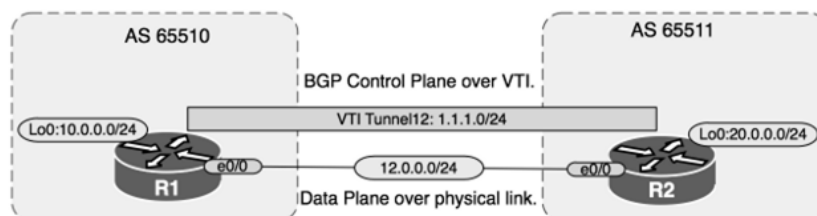


图 7.8-3

## 7.8.2 Azure 端 BGP VPN 的相关配置

### 1. 登录账户

```
Add-AzureRmAccount -EnvironmentName AzureChinaCloud
```

### 2. 查看账户下的订阅信息

```
Get-AzureRmSubscription | sort SubscriptionName,SubscriptionId | select SubscriptionName,SubscriptionId
```

### 3. 指定配置 BGP VPN 的订阅

```
$SubID = '26f0f62d-79b0-4f16-a276-xxxxxxxxxxxx'
Select-AzureRmSubscription -SubscriptionId $SubID
```

### 4. 声明创建 BGP VPN 时所使用的变量信息

```
$RG1 = "jackBGPRG1"
$Location1 = "China North"
$VNetName1 = "jackBGPVNet1"
$FESubName1 = "FrontEnd"
$BESubName1 = "Backend"
$GWSubName1 = "GatewaySubnet"
$VNetPrefix11 = "10.11.0.0/16"
$VNetPrefix12 = "10.12.0.0/16"
$FESubPrefix1 = "10.11.0.0/24"
$BESubPrefix1 = "10.12.0.0/24"
$GWSubPrefix1 = "10.12.255.0/27"
$VNet1ASN = 65010
$DNS1 = "119.29.29.29"
$GWName1 = "jackBGPVNet1GW"
$GWIPName1 = "jackBGPVNet1GWIP"
$GWIPconfName1 = "jackBGPgwipconf1"
```

### 5. 创建资源组

```
New-AzureRmResourceGroup -Name $RG1 -Location $Location1
```

### 6. 在资源组中创建虚拟网络

```
$fesub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName1
-AddressPrefix $FESubPrefix1
$besub1 = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName1
-AddressPrefix $BESubPrefix1
$gwsb1 = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName1
-AddressPrefix $GWSubPrefix1

New-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName $RG1
-Location $Location1 -AddressPrefix $VNetPrefix11,$VNetPrefix12 -Subnet
$fesub1,$besub1,$gwsb1
```

### 7. 创建动态 IPv4 地址作为 VPN 网关

```
$gwip1 = New-AzureRmPublicIpAddress -Name $GWIPName1 -ResourceGroupName
$RG1 -Location $Location1 -AllocationMethod Dynamic

$vn1 = Get-AzureRmVirtualNetwork -Name $VNetName1 -ResourceGroupName
$RG1
$subnet1 = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet "
-VirtualNetwork $vn1
$gwipconf1 = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName1
-Subnet $subnet1 -PublicIpAddress $gwip1
```

## 8. 创建 VPN 网关调用之前创建的 IPv4 地址

```
New-AzureRmVirtualNetworkGateway -Name $GWName1 -ResourceGroupName $RG1
-Location $Location1 -IpConfigurations $gwipconf1 -GatewayType Vpn -VpnType
RouteBased -GatewaySku HighPerformance -Asn $VNet1ASN
```

## 9. 查看 VPN 网关的 BGP 相关信息

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1
-ResourceGroupName $RG1
$vnet1gw.BgpSettingsText
{
    "Asn": 65010, //本端 VPN Gateway ASN Num
    "BgpPeeringAddress": "10.12.255.30", //本端 BGP Peer 地址
    "PeerWeight": 0
}
```

## 10. 链接本地 VPN，声明链接本地 VPN 所需的参数

```
$RG5 = "jackLocalBGPRG"
$Location5 = "China North"
$LNGName5 = "ConntoLocal"
$LNGPrefix50 = "10.52.255.254/32"
$LNGIP5 = "106.120.78.190"
$LNGASN5 = 65050
$BGPPeerIP5 = "10.52.255.254"
```

## 11. 创建用户本地 VPN 的资源组，也可与虚拟网络在相同资源组

```
New-AzureRmResourceGroup -Name $RG5 -Location $Location5
```

## 12. 创建本地网络网关(本地 VPN 设备的相关信息)

```
New-AzureRmLocalNetworkGateway -Name $LNGName5 -ResourceGroupName $RG5
-Location $Location5 -GatewayIpAddress $LNGIP5 -AddressPrefix $LNGPrefix50 -Asn
$LNGASN5 -BgpPeeringAddress $BGPPeerIP5
```

## 13. 获取 VNET VPN Gateway 网关和本地网络网关信息并指定链接名称

```
$vnet1gw = Get-AzureRmVirtualNetworkGateway -Name $GWName1
-ResourceGroupName $RG1
$lng5gw = Get-AzureRmLocalNetworkGateway -Name $LNGName5
-ResourceGroupName $RG5
$Connection15 = 'ConntoLocalSite'
```

## 14. 创建链接

```
New-AzureRmVirtualNetworkGatewayConnection -Name $Connection15
-ResourceGroupName $RG1 -VirtualNetworkGateway1 $vnet1gw -LocalNetworkGateway2
$lng5gw -Location $Location1 -ConnectionType IPsec -SharedKey 'AzureA1b2C3'
-EnableBGP $True
```

## 15. 如创建链接时报以下错误，建议您升级 AzurePowerShell 版本

```

WARNING: The output object type of this cmdlet will be modified in a future
release.
New-AzureRmVirtualNetworkGatewayConnection :
Mapping types:
UInt64 -> Nullable`1
System.UInt64 -> System.Nullable`1[[System.Int64, mscorlib,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]
Destination path:
VirtualNetworkGatewayConnection.EgressBytesTransferred.EgressBytesTransf
erred
Source value:
0
At line:1 char:1
+ New-AzureRmVirtualNetworkGatewayConnection -Name azure-to-onprem-tu ...

```

## 16. 如在执行 PowerShell 命令时遇到报错信息可参考以下链接尝试解决:

<https://www.azure.cn/documentation/articles/resource-manager-common-deployment-errors/>

## 7.8.3 在本地配置 IPsec VPN（以 Cisco CSR 1000v 为例）

!设置 IKEv2 初期阶段两次交互(IKE\_SA\_INIT, IKE AUTH)需要的相关信息(用什么方式认证 Azure 目前只支持预共享密钥方式认证, 加密算法, 哈希算法, DH 组,)

```

crypto ikev2 proposal toazurecn
  encryption aes-cbc-256 aes-cbc-128 3des
  integrity sha1
  group 2
  exit

crypto ikev2 policy toazurecn-policy
  proposal toazurecn
  exit

crypto ikev2 keyring toazurecn-keyring
  peer 139.219.194.42
  address 139.219.194.42
  pre-shared-key AzureAlb2C3
  exit
  exit

crypto ikev2 profile toazurecn-profile
  match address local interface GigabitEthernet1
  match identity remote address 139.219.194.42 255.255.255.255
  authentication local pre-share
  authentication remote pre-share
  keyring local toazurecn-keyring
  exit

```



```

!设置第二阶段 CREATE_CHILD_SA 协商所使用的加密, 认证, VPN 模式信息
crypto ipsec transform-set toazurecn-trans esp-aes 256 esp-sha-hmac
mode tunnel
exit

crypto ipsec profile toazurecn-pro
set transform-set toazurecn-trans
set ikev2-profile toazurecn-profile
exit

!设置 SVTI 接口信息, 该接口只能为 169.254.x.x 网段地址, 如需设置多条 VPN Tunnel 可为
每条 Tunnel 设置不同网段 IP, 例如: Tunnel1: 169.254.0.1/24, Tunnel2: 169.254.1.1
Azure 端为相同网段随机 IP.
interface Tunnel1
ip address 169.254.1.1 255.255.255.0
ip tcp adjust-mss 1350
tunnel source GigabitEthernet1
tunnel mode ipsec ipv4
tunnel destination 139.219.194.42
tunnel protection ipsec profile toazurecn-pro
exit

!配置 BGP 控制层面流量走 VTI
ip route 10.12.255.30 255.255.255.255 Tunnel1

!BGP 配置, 仅用于连通性测试
router bgp 65050
bgp router-id 10.52.255.254 !本例中该地址为本地 VPN 设备 loopback 口地址
network 10.2.1.0 mask 255.255.255.0
neighbor 10.12.255.30 remote-as 65011。
neighbor 10.12.255.30 ebgp-multihop 2

```

#### 7.8.4 在 Azure 端使用 PowerShell 查看链接情况

```

Get-AzureRmVirtualNetworkGatewayConnection -Name $Connection15 -Resource
GroupName $RG
Name                : conntojacknVPN //链接名称
ResourceGroupName   : jackERG        //链接所在的资源组
Location             : chinaeast     //链接所在的地域
SharedKey            : Cisco123      //链接的密钥
ConnectionStatus     : Connected     //链接的状态
EgressBytesTransferred : 1464         //VPN 网关出口流量
IngressBytesTransferred : 1641       //VPN 网关入口流量

```

# 第八章 安全配置

对于公有云来说，用户最关心的当属安全方面的问题了。本章重点围绕安全方面的配置和注意事项进行了详细介绍，包括限制公网访问的访问控制列表（ACL，Access Control List）和网络安全组（NSG，Network Security Group）的配置以及案例分析，此外，对于在 Azure 资源管理器模型中引入的基于角色的访问控制（RBAC，Role-Based Access Control）也进行了深入的讨论和案例分析。

## 8.1 访问控制列表（ACL）

### 8.1.1 什么是终结点访问控制列表（ACL）

终结点访问控制列表（ACL）是可用于 Azure 部署的安全增强。利用 ACL，你可以选择允许还是拒绝虚拟机终结点的流量。此数据包筛选功能额外提供了一层安全性。只能为终结点指定网络 ACL，无法为虚拟网络或虚拟网络中包含的特定子网指定 ACL。

提醒注意：建议尽可能使用网络安全组（NSG），而不要使用 ACL。

使用网络 ACL 可以实现以下目的：

- 根据远程子网 IPv4 地址范围选择允许还是拒绝传入流量流向虚拟机输入终结点。
- 方块列表 IP 地址。
- 为每个虚拟机终结点创建多个规则。
- 为每个虚拟机终结点指定最多 50 个 ACL 规则。
- 使用规则排序可确保将一组正确的规则应用于给定的虚拟机终结点（最低到最高）。
- 为特定远程子网 IPv4 地址指定 ACL。

### 8.1.2 ACL 的工作原理

（1）ACL 是包含规则列表的对象。当你创建 ACL 并将其应用于虚拟机终结点时，数据包筛选将在 VM 的主机节点上发生。这意味着，来自远程 IP 地址的流量将通过主机节点筛选以匹配 ACL 规则，而不是在你的 VM 上进行筛选。这可以防止 VM 将宝贵的 CPU 周期耗费在数据包筛选上。

（2）创建虚拟机时，将会设置一个默认 ACL 来阻止所有传入流量。但是，如果创建了一个终结点（针对端口 3389），则会修改默认 ACL 以允许该终结点的所有入站流量。随后，将允许来自任何远程子网的传入流量流向该终结点，而且不需要配置防火墙。除非为这些端口创建了终结点，否则将阻止所有其他端口的传入流量。默认情况下允许出站流量。

8.1.3 客户案例

想实现 XXXX.chinacloudapp.cn:3389 这个 cloud service 能够被虚拟网络和本地网络访问，不能够被其他 Internet 访问，该如何设置 ACL？

(1) 具体配置截图，进行虚拟机选项，终结点”页将列出该虚拟机的所有当前终结点。（此示例中的是 Windows VM。如果是 Linux VM，则默认显示一个 SSH 终结点。）请参考图 8.1-1。

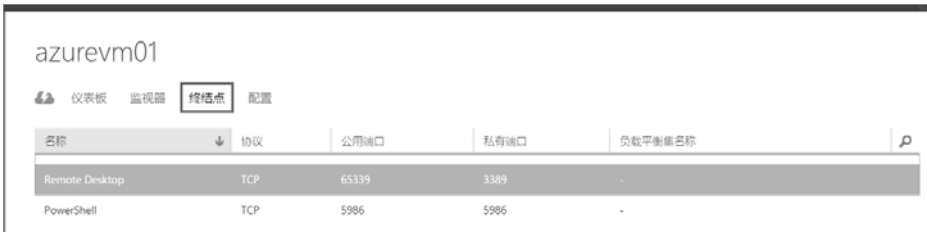


图 8.1-1

(2) 如何实现需求:进入 Remote Desktop，进行如下配置即可,填入本地网络公网 IP 地址（段）65.0.0.0/8（32）请参考图 8.1-2。



图 8.1-2

网络 ACL 将采用越小越优先的规则顺序。如果指定某一个 IP 拒绝访问，可以写成截图显示内容：175.1.0.1/32。请参考图 8.1-3。



图 8.1-3

## 8.2 网络安全组（NSG）

### 8.2.1 什么是网络安全组（NSG）

网络安全组（NSG）包含一系列访问控制列表（ACL）规则，这些规则可以允许或拒绝虚拟网络中流向 VM 实例的网络流量。NSG 可以与子网或该子网中的各个 VM 实例相关联。当 NSG 与某个子网相关联时，ACL 规则适用于该子网中的所有 VM 实例。另外，可以进一步通过将 NSG 直接关联到单个 VM 对流向该 VM 的流量进行限制。

NSG 包含两种类型的规则：进站规则和出站规则。在每组中，规则的优先级必须保持唯一。请参考图 8.2-1。



图 8.2-1

### 8.2.2 默认标记

默认标记是系统提供的针对某类 IP 地址的标识符。你可以使用任何规则的源地址前缀和目标地址前缀属性中的默认标记。有三个可使用的默认标记。

- VIRTUAL\_NETWORK: 此默认标记表示你的所有网络地址空间。它包括虚拟网络地址空间（Azure 中定义的 CIDR 范围）以及所有连接的本地地址空间和连接的 Azure VNet（本地网络）。
- AZURE\_LOADBALANCER: 此默认标记表示 Azure 的基础结构负载均衡器。这将转换到 Azure 数据中心 IP，从中进行 Azure 的运行状况探测。
- INTERNET: 此默认标记表示虚拟网络外部的 IP 地址空间，可以通过公共 Internet 进行访问。此范围还包括 Azure 拥有的公共 IP 空间。

### 8.2.3 客户案例

如何限制虚拟网络的一台计算机不能访问向外访问 Internet，需要正常远程 RDP。（此示例中的第一部分是经典模式下 Windows VM 使用 PowerShell 完成配置，第二部分是新门户中使用 GUI 界面完成配置）如图 8.2-2 所示。

#### 1. 创建一个安全组

```
PS C:\> New-AzureNetworkSecurityGroup -Name "BlockInternet" -Location "China East "
```

#### 2. 定义变量

```
PS C:\> $NSGGroup = Get-AzureNetworkSecurityGroup -Name BlockInternet
```

#### 3. 配置针对 10.20.0.4 这台虚拟机 Outbound，拒绝访问 Internet

```
PS C:\> $NSGGroup | Set-AzureNetworkSecurityRule -Name block-internet -Action Deny -Protocol * -Type Outbound -Priority 200 -SourceAddressPrefix '10.20.0.4/32' -SourcePortRange * -DestinationAddressPrefix Internet -DestinationPortRange *
```

#### 4. 配置针对 10.20.0.4 这台虚拟机 Inbound 规则,允许 Inbound

```
PS C:\> $NSGGroup | Set-AzureNetworkSecurityRule -Name Allow-inbound -Action Allow -Protocol * -Type Inbound -Priority 200 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix '10.20.0.4/32' -DestinationPortRange *
```

#### 5. 将规则应用在虚拟机所在子网

```
$NSGGroup | Set-AzureNetworkSecurityGroupToSubnet -VirtualNetworkName azurevnet -SubnetName Subnet-1
```

#### 6. 其它命令参考

##### 1) 取消分配

```
Remove-AzureNetworkSecurityGroupAssociation -VirtualNetworkName azurevnet -SubnetName Subnet-1
```

##### 2) 删除 NSG

```
Remove-AzureNetworkSecurityGroup
```

Libza Portal 创建新网络安全组，请参考图 8.2-2。

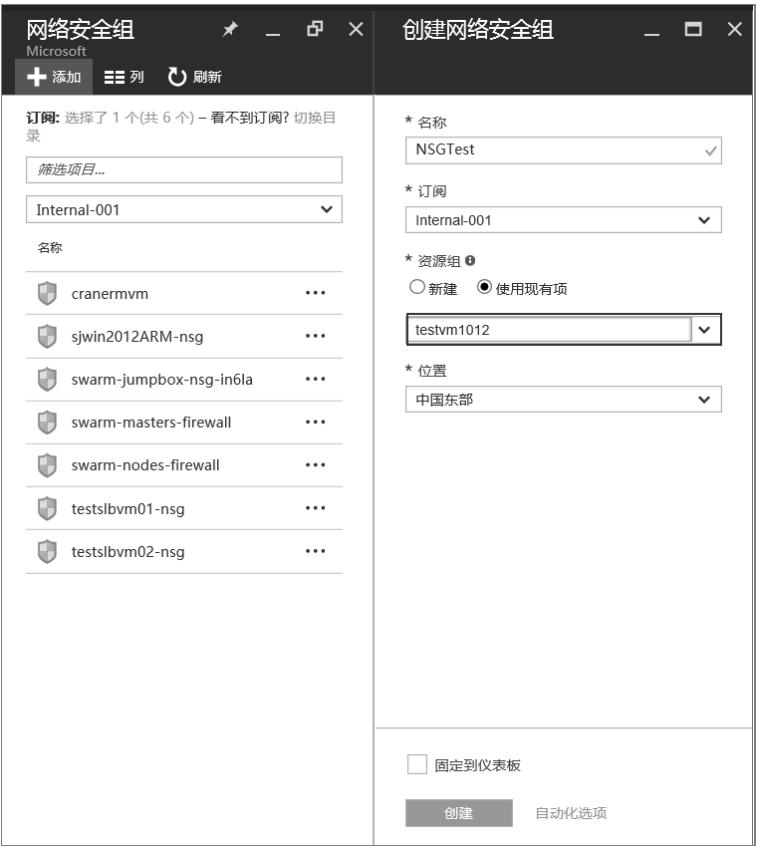


图 8.2-2

添加入站规则允许 3389，出现规则拒绝访问 Internet。请参考图 8.2-3。

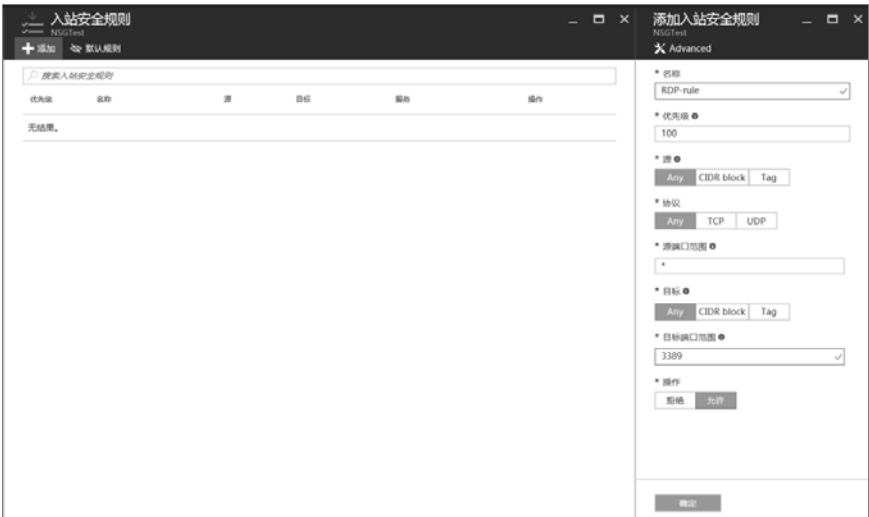


图 8.2-3

请参考图 8.2-4。

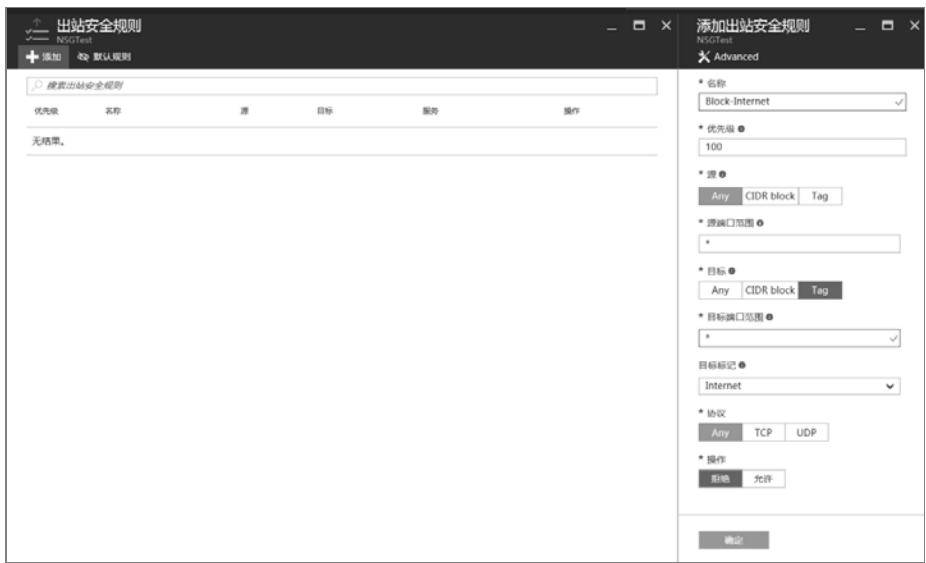


图 8.2-4

查看规则，通过远程桌面可以连接 Azure VM，80 端口正常访问，虚拟机无法访问 Internet。请参考图 8.2-5。

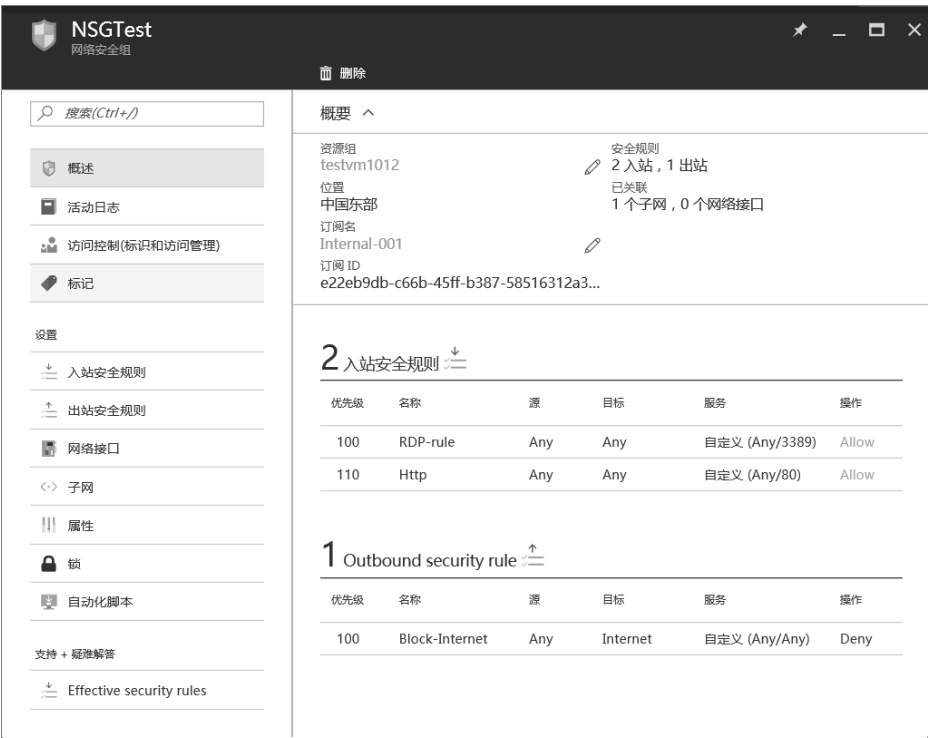


图 8.2-5

将网络安全组关联给虚拟网络或具体的网络接口（即指定 VM）请参考图 8.2-6。

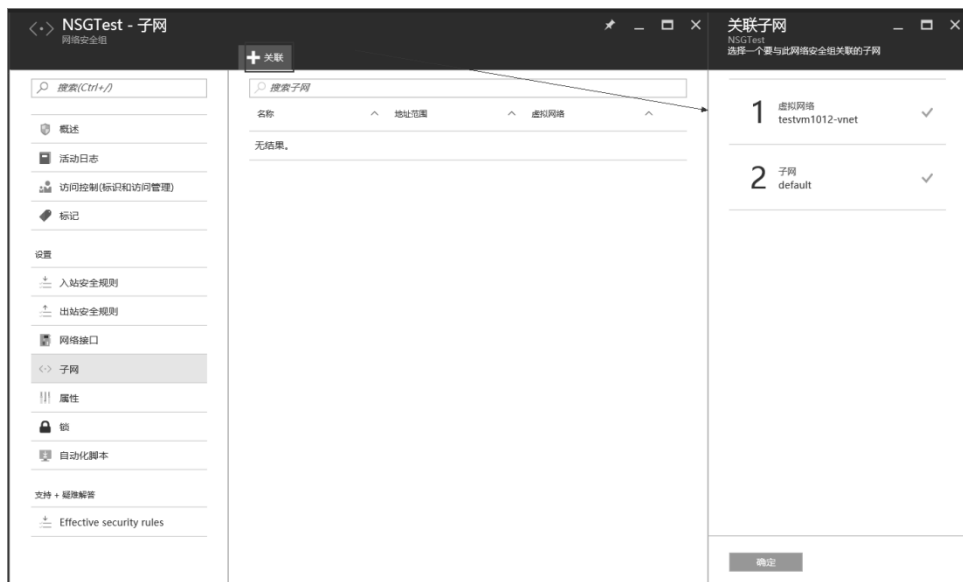


图 8.2-6

帮助排错：

```
get-help Set-AzureRmVirtualNetworkSubnetConfig -Detailed
```

## 8.2.4 注意事项规则需要避开的项目

(1) 网关子网：不要将 NSG 关联到网关子网。

Azure 使用称为“网关”子网的特殊子网处理其他 VNet 和本地网络的 VPN 网关。将 NSG 关联到此子网将导致 VPN 网关停止按预期方式正常工作。

(2) 主机节点的虚拟 IP 地址 168.63.129.16：不要限制此 IP 的流量

基本基础结构服务（例如 DHCP、DNS 和运行状况监视）是通过虚拟化主机 IP 地址 168.63.129.16 提供的。此公用 IP 地址属于 Microsoft，并将是唯一的用于所有区域的虚拟化 IP 地址，而且没有其他用途。此 IP 地址映射到托管虚拟机的服务器计算机（主机节点）的物理 IP 地址。主机节点充当 DHCP 中继、DNS 递归解析器，以及进行负载均衡器运行状况探测和计算机运行状况探测的探测源。不应将针对此 IP 地址的通信视为一种攻击。

(3) 出站端口 1688：不要限制此端口的流量

许可（密钥管理服务）：在虚拟机中运行的 Windows 映像应该获得许可。因此，将会向处理此类查询的密钥管理服务主机服务器发送许可请求。这将始终在出站端口 1688 上进行。

(4) 正常进行 IaaS VM 备份的 NSG 规则，备份扩展需要访问公共 Internet 才能工作。无法访问公共 Internet 时，扩展安装、备份操作、备份状态都可能失败。所以要将 Azure 数据中心 IP 范围加入允许列表，为 HTTP 流量创建路径。



## 8.3 基于角色的访问控制（RBAC）

### 8.3.1 什么是基于角色的访问控制

Azure 基于角色的访问控制（RBAC）可用于对 Azure 进行细致的访问管理。使用 RBAC，你可以在开发运营团队中对职责进行分配，仅向用户授予执行作业所需的访问权限。你可以仅授予用户执行其作业所需的访问次数。

RBAC 是 Role Based Access Control 是基于角色的接入控制的简称。在 Azure 推出 ARM 以后，对 Azure 各种资源的管理粒度已经非常细致，使得 RBAC 成为可能。请参考图 8.3-1。



图 8.3-1

通过 RBAC 可以非常方便的给不同的用户分配不同的资源的不同权限。请参考图 8.3-2。

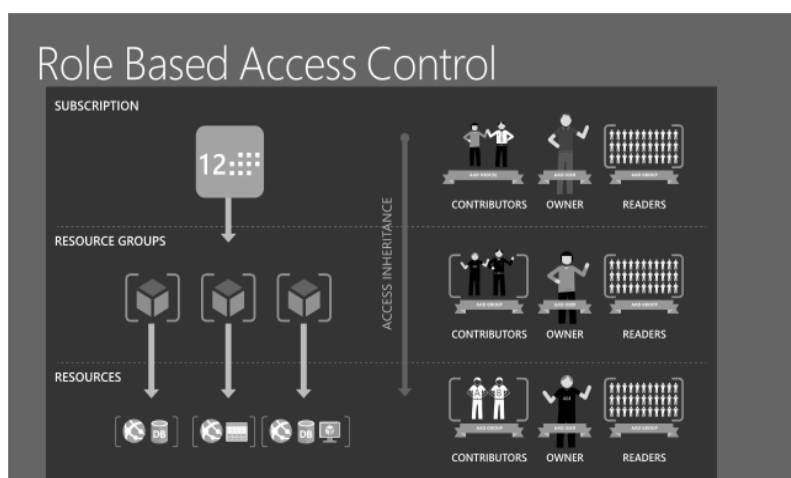


图 8.3-2

内置角色

Azure RBAC 有三种适用于所有资源类型的基本角色：

**所有者**具有对所有资源的完全访问权限，包括委派对其他用户的访问权限。

**参与者**可以创建和管理所有类型的 Azure 资源，但不能将访问权限授予其他用户。

**读者**可以查看现有的 Azure 资源。

### 8.3.2 注意事项

Azure 中的其他 RBAC 角色允许对特定的 Azure 资源进行管理。例如，虚拟机参与者角色允许用户创建和管理虚拟机。它并不授予其访问虚拟机连接的虚拟网络或子网的权限。

经典订阅管理员和共同管理员具有对 Azure 订阅的完全访问权限。

Azure RBAC 仅支持 Azure 门户和 Azure Resource Manager API 中的 Azure 资源的管理操作。并不是 Azure 资源的所有数据级别操作都可通过 RBAC 授权。例如，可以使用 RBAC 对存储账户进行管理，但是不能使用 RBAC 管理存储账户中的 blob 或表。同样，可以管理 SQL 数据库，但是不能管理其中的表。

### 8.3.3 客户案例

如何配置让用户只能访问指定虚拟机？

如果创建用户时没有分配订单和协同管理员，是无法登录经典管理门户的。所以这个需求在经典管理门户中并不能实现。请参考图 8.3-3。



图 8.3-3

Libza Portal 配置如下：

使用服务管理员账号查看所有资源，可以看到资源很多：请参考图 8.3-4。

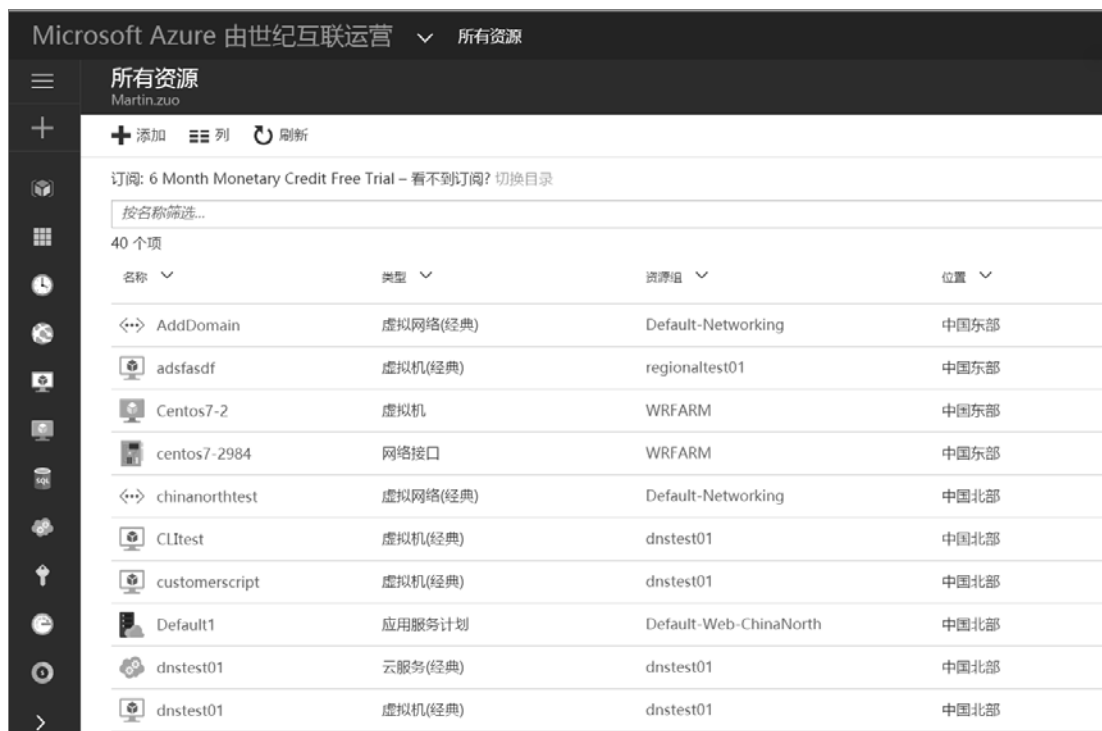


图 8.3-4

接下为我们为 RBACtest 用户分配可以查看指定虚拟机的权限。请参考图 8.3-5。

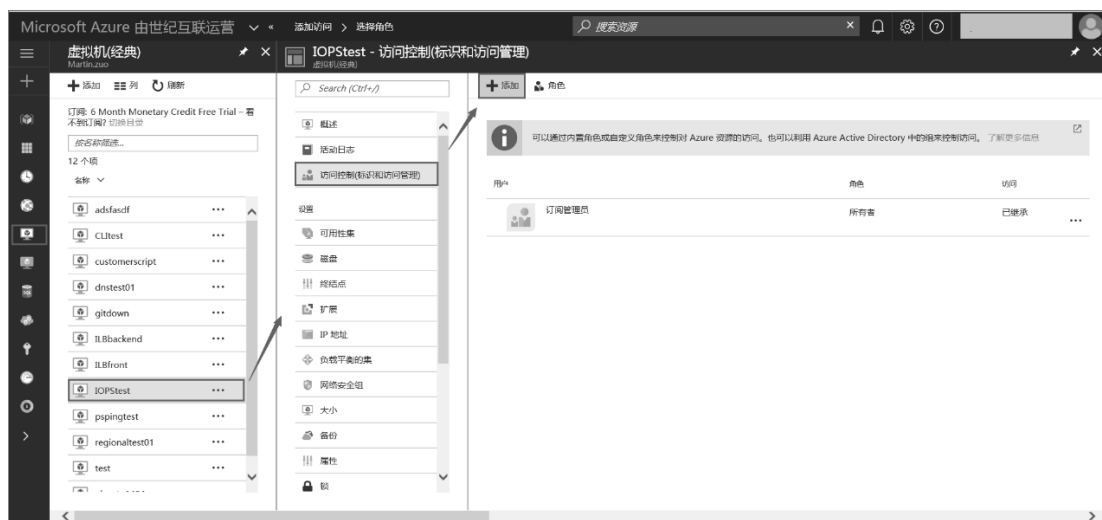


图 8.3-5

选择之前在经典管理门户中创建的用户，赋予“读者”权限。请参考图 8.3-6。

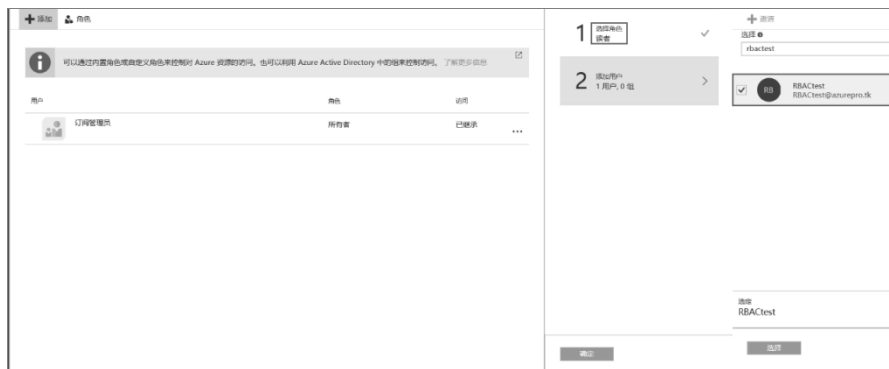


图 8.3-6

RBACtest 登录验证，可以看到当前虚拟机处于关闭状态，用户无权限启动。请参考图 8.3-7。

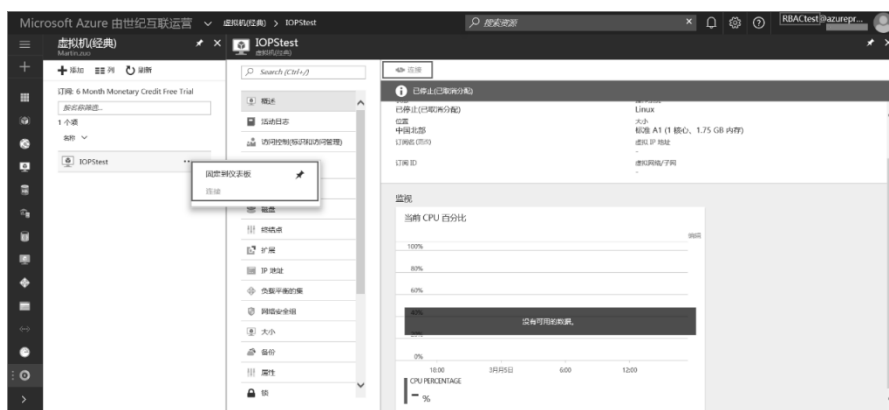


图 8.3-7

RBACtest 赋予“参与者”权限后，登录验证，可以看到针对当前虚拟机进行的权限已经增加，允许用户管理虚拟机。请参考图 8.3-8。

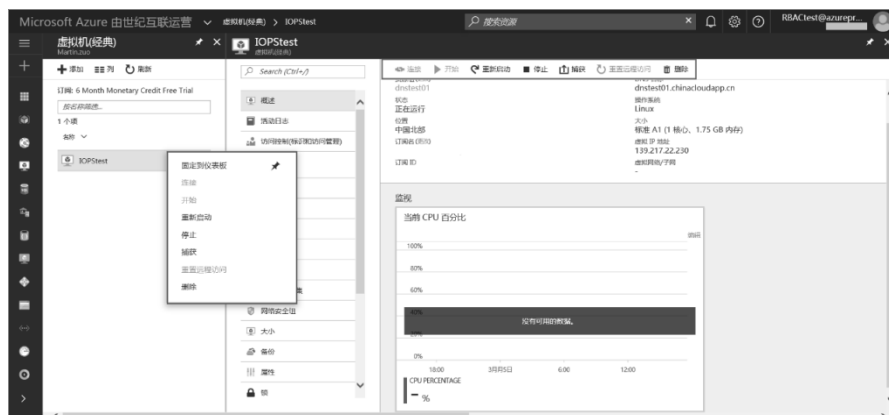


图 8.3-8

## 8.4 Microsoft Antimalware

Microsoft Antimalware for Azure Cloud Services and Virtual Machines 提供实时的保护功能，有助于识别和移除病毒、间谍软件及其他恶意软件，可在已知恶意或垃圾软件试图在系统上自动安装或运行时发出可配置的警报。

### 8.4.1 经典管理门户配置

从 Azure 门户创建新的 VM 时，可启用 Microsoft Antimalware 解决方案。请参考图 8.4-1。



图 8.4-1

要在现有虚拟机上安装，必须运行以下 cmdlet。

获取 VM：

```
$vm = Get-AzureVM -ServiceName $servicename -Name $name
```

将 Microsoft Antimalware 代理添加到虚拟机

```
Set-AzureVMExtension -Publisher Microsoft.Azure.Security -ExtensionName  
IaaSAntimalware -Version 1.* -VM $vm.VM
```

更新要安装反恶意软件代理的 VM：

```
Update-AzureVM -Name $servicename -ServiceName $name -VM $vm.VM
```

卸载扩展程序：

可以使用以下 cmdlet 卸载上述添加的扩展程序。

获取 VM：

```
$vm = Get-AzureVM -ServiceName $servicename -Name $name
```

从虚拟机卸载扩展程序：

```
Set-AzureVMExtension -Publisher $publishername -ExtensionName  
$extensionname -Version $version -VM $vm.VM -Uninstall
```

更新要卸载扩展程序的 VM：

```
Update-AzureVM -Name $servicename -ServiceName $name -VM $vm.VM
```

这种集成将 Azure 虚拟机和现有安全解决方案相结合，使其便于共同部署和管理

## 8.4.2 Libza Portal 模式 ARM 虚拟机配置

New Portal 上创建 VM 的时候看不到开启 Extension 的 UI 选项，所以需要在创建完 VM 之后通过 PowerShell 命令开启该扩展，

### 1. ARM 虚拟机加载 Antimalware 扩展

1) 通过 PowerShell 加载 Antimalware 扩展，应该至少先定义一个 Antimalware 启动的配置否则无法加载成功：

```
$SettingsString=@{ "AntimalwareEnabled" = "true" }  
Set-AzureRmVMExtension -Publisher Microsoft.Azure.Security -ExtensionType  
IaaSAntimalware -ResourceGroupName yourgroupname -VMName youvmname -Name  
IaaSAntimalware -TypeHandlerVersion 1.3 -Location chinanorth -Settings  
$SettingsString
```

### 2. 关于如何获取最新的 ARM 虚拟机 Antimalware 镜像扩展信息的方法

1) 确认最新的 Antimalware 服务发布者的名称（有的时候这个名字会因为平台的更新改变）：

```
Get-AzureRmVMImagePublisher -Location chinanorth
```

2) 获取镜像类型信息：

```
Get-AzureRmVMExtensionImageType -PublisherName Microsoft.Azure.Security  
-Location chinanorth
```

3) 获取镜像版本信息：

```
$PublisherName = 'Microsoft.Azure.Security'  
$PublisherType = 'IaaSAntimalware'  
$Location = 'chinanorth'  
$Publisher = Get-AzureRmVMExtensionImage -PublisherName $PublisherName  
-Type $PublisherType -Location $Location
```

```
$Version = $Publisher.Version.Substring(0,3)
$Version
```

### 8.4.3 用户案例

创建了 VM 并安装了 Antimalware, 进入 VM 无法打开 system center endpoint protection 程序。

**原因：**默认为禁用

**启用方法：**进入指定目录执行以下命令：

```
C:\Program Files\Microsoft Security Client>ConfigSecurityPolicy.exe
cleanuppolicy.xml
```

请参考图 8.4-2。

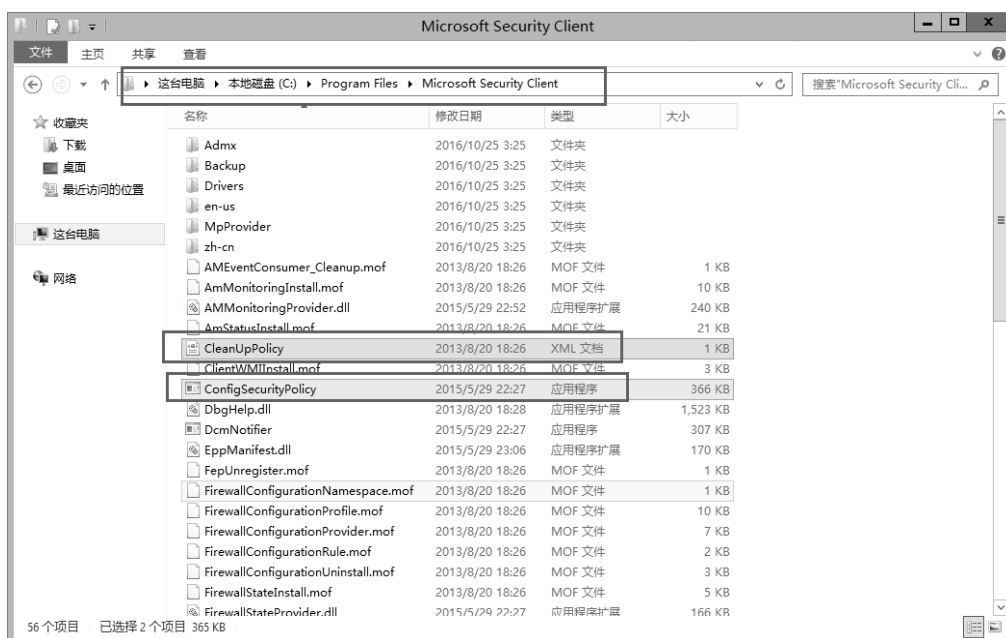


图 8.4-2

执行操作，请参考图 8.4-3。

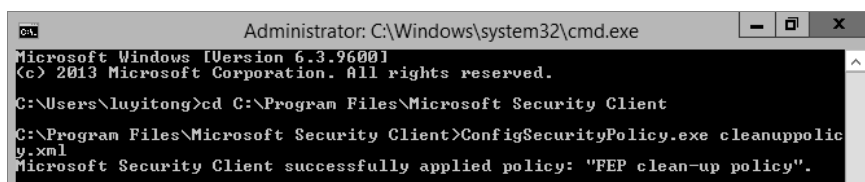


图 8.4-3

验证结果，请参考图 8.4-4。

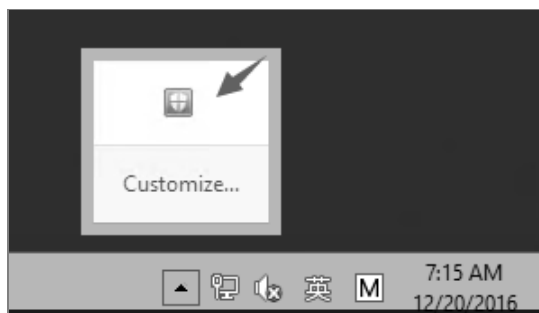


图 8.4-4

请参考图 8.4-5。

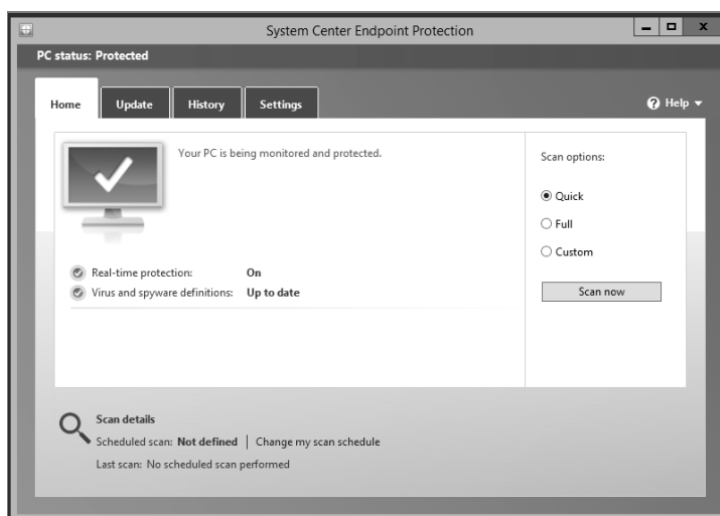


图 8.4-5



## 第九章 负载均衡与高可用设计

随着越来越多的用户将应用部署到 Azure 中，来自终端用户的访问量越来越大，应用的性能和可用性也渐渐称为 Azure 用户关心的一个重要方面。本章针对各种不同的负载均衡方法以及可用性进行了深入讨论，从原生的内/外部负载均衡，到基于第七层协议进行负载均衡的应用程序网关，还有可用性集的设计，内容中融入了大量实际案例，相信对于读者部署实际环境会大有帮助。

### 9.1 面向 Internet 的负载均衡

Azure load balancer 是位于第 4 层 (TCP, UDP) 的负载均衡器。该负载均衡器可以在云服务或负载均衡器集的虚拟机中运行状况良好的服务实例之间分配传入流量，从而提供高可用性。Azure Load Balancer 还可以在多个端口和/或多个 IP 地址上显示这些服务，如图 9.1-1 所示。

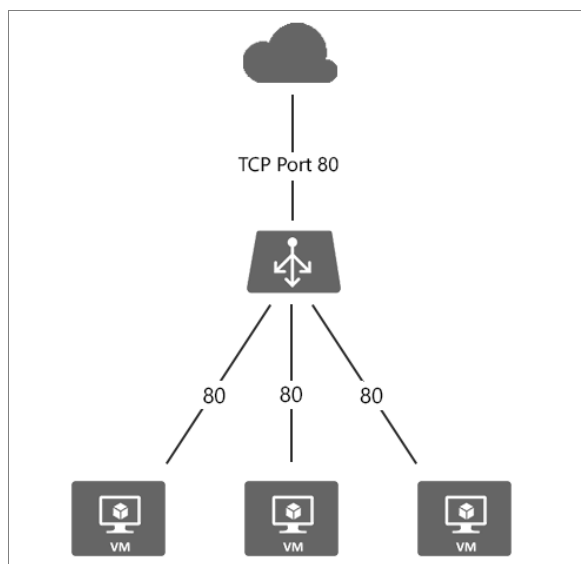


图 9.1-1

可以将负载均衡器配置为用于：

- 对传入到虚拟机（VM）的 Internet 流量进行平衡负载。我们将此方案中的负载均衡器作为一个面向 Internet 的负载均衡器。
- 对虚拟网络（VNet）和云服务中 VM 之间的流量或本地计算机和跨界虚拟网络中

VM 之间的流量进行平衡负载。我们将此方案中的负载均衡器作为一个内部负载均衡器（ILB）。

- 将外部流量转发到特定的 VM 实例。

当 Internet 客户端将网页请求发送到 TCP 端口 80/443 上的云服务的公共 IP 地址时，Azure Load Balancer 会在负载均衡集中的三个虚拟机之间分发请求。

默认情况下，Azure Load Balancer 在多个虚拟机实例之间平均分发网络流量。另外，还可以配置会话关联。

### 9.1.1 创建面向 Internet 的负载均衡器

需要实现的目标（资源管理器部署模型）：

- 在端口 80 上创建一个接收网络流量的负载均衡器，并将负载均衡流量发送到虚拟机“lytWeb1”和“lytWeb2”。
- 在负载均衡器后面创建虚拟机的远程桌面访问/SSH 的 NAT 规则。
- 创建运行状况探测，如图 9.1-2 所示。

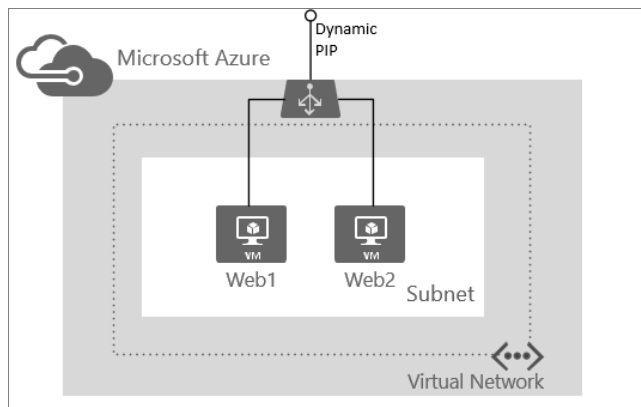


图 9.1-2

若要创建 Azure 虚拟机的负载平衡集，请使用以下步骤：

- 步骤 1：同一云服务下创建两台虚拟机。
- 步骤 2：使用第一台虚拟机创建负载均衡集。
- 步骤 3：将第二台虚拟机添加到负载均衡集。

#### 9.1.1.1 创建虚拟机

##### 1. 创建第一台 VM

登录到 Azure 经典管理门户。你可以使用“从库中”或“快速创建”方法创建第一台虚拟机（lytwin01）。

##### 2. 在同一云服务中创建第二台 VM

使用“从库中”方法，在第一台虚拟机的同一云服务中创建第二台虚拟机（lytwin02），如图 9.1-3 所示。

实例 映像 磁盘					
名称	状态	订阅	位置	DNS 名称	
lytwin01	✔ 正在运行	WATSTest03	中国北部	lytcloud.chinacloudapp.cn	
lytwin02	✔ 正在运行	WATSTest03	中国北部	lytcloud.chinacloudapp.cn	

图 9.1-3

9.1.1.2 使用虚拟机创建负载均衡集

- (1) 在 Azure 管理门户中，单击“虚拟机”，然后单击第一台虚拟机的名称。
- (2) 单击“终结点”，然后单击“添加”。
- (3) 在“将终结点添加到虚拟机”页上，单击下一步。
- (4) 在“指定终结点的详细信息”页上执行以下操作，如图 9.1-4 所示。
  - 在“名称”中，键入终结点的名称，并从通用协议的预定义终结点列表中进行选择。
  - 在“协议”中，根据需要选择终结点类型需要的协议，例如 TCP 或 UDP。
  - 在“公用端口”和“专用端口”中，根据需要键入你希望虚拟机使用的端口号。你可以使用虚拟机上的专用端口和防火墙规则，从而以适合你的应用程序的方式重定向流量。专用端口可以与公用端口一样。例如，对于 Web (HTTP) 流量的终结点，你可将端口 80 指定为公用端口和专用端口。

添加终结点

指定终结点的详细信息

名称

HTTP

协议

TCP

公用端口

80

私有端口

80

☒ 创建负载均衡集

☐ 启用直接服务器返回

←

→

图 9.1-4

- (5) 选择“创建负载均衡集”，然后单击下一步。
- (6) 在“配置负载均衡集”页上，键入负载均衡集的名称，然后分配用于 Azure 负载均衡器的探测行为的值。负载均衡器使用探测来确定负载均衡集中的虚拟机是否可用于接收传入流量，如图 9.1-5 所示。
- (7) 单击复选标记以创建负载均衡的终结点。你将在虚拟机的“终结点”页的“负载

平衡集名称”列中看到“是”。



图 9.1-5

9.1.1.3 将虚拟机添加到负载均衡集

创建负载均衡集之后，将其他虚拟机添加到其中。对同一云服务中的每个虚拟机执行以下操作。

- (1) 在管理门户中，依次单击“虚拟机”、虚拟机的名称、“终结点”、“添加”。
- (2) 在“将终结点添加到虚拟机”页上，单击“将终结点添加到现有负载均衡集”，选择负载均衡集的名称，然后单击下一步，如图 9.1-6 所示。



图 9.1-6

(3) 在“指定终结点详细信息”页上，键入终结点的名称，然后单击复选标记，如图 9.1-7 所示。

图 9.1-7

#### 9.1.1.4 验证负载均衡功能

使用 IE 访问云服务域名：<http://lytcloud.chinacloudapp.cn>，刷新页面会循环访问负载均衡集中的两台 Web Server，如图 9.1-8 所示。



图 9.1-8

#### 9.1.1.5 为负载均衡器配置空闲 TCP 超时设置

##### 1. 将实例级公共 IP 的 TCP 超时值配置为 15 分钟

```
Set-AzurePublicIP -PublicIPName Webip -VM MyVM -IdleTimeoutInMinutes 15
```

备注：IdleTimeoutInMinutes 是可选项。如果未设置，默认超时为 4 分钟。可接受的超时范围为 4 到 30 分钟。

## 2. 在虚拟机上创建 Azure 终结点时设置空闲超时

(1) 更改终结点的超时设置，请执行以下命令：

```
Get-AzureVM -ServiceName "mySvc" -Name "MyVM" | Add-AzureEndpoint -Name
"HttpIn" -Protocol "tcp" -PublicPort 80 -LocalPort 80 -IdleTimeoutInMinutes
15 | Update-AzureVM
```

(2) 检索空闲超时配置，请执行以下命令：

```
PS C:\> Get-AzureVM -ServiceName " MyService " -Name " MyVM " |
Get-AzureEndpoint
VERBOSE: 5:23:50 PM - Completed Operation: Get Deployment
LBSetName : HttpLoadBalance
LocalPort : 80
Name : HTTP
Port : 80
Protocol : tcp
Vip : 65.52.xxx.xxx
ProbePath :
ProbePort : 80
ProbeProtocol : tcp
ProbeIntervalInSeconds : 15
ProbeTimeoutInSeconds : 31
EnableDirectServerReturn : False
Acl : {}
InternalLoadBalancerName :
IdleTimeoutInMinutes : 15
```

## 3. 在负载均衡的终结点集上设置 TCP 超时

如果终结点是负载均衡的终结点集的一部分，则必须在负载均衡的终结点集上设置 TCP 超时。例如：

```
Set-AzureLoadBalancedEndpoint -ServiceName " MyService " -LBSetName "
HttpLoadBalance " -Protocol tcp -LocalPort 80 -ProbeProtocolTCP -ProbePort 80
-IdleTimeoutInMinutes 15
```

## 4. 更改云服务的超时设置

可以使用 Azure SDK 来更新云服务。在 .csdef 文件中进行云服务的终结点设置。更新云服务部署的 TCP 超时需要升级部署。例外情况是如果仅针对公共 IP 指定 TCP 超时，则无需升级。公共 IP 设置位于 .cscfg 文件中，可以通过部署更新和升级进行更新。

(1) 终结点设置的.csdef 更改如下：

```
<WorkerRole name= " worker-role-name " vmsize= " worker-role-size "
enableNativeCodeExecution= " [true|false] " >
```

```

    <Endpoints>
      <InputEndpoint name= "input-endpoint-name" protocol= "[http|https|tcp|udp]"
localPort= " local-port-number " port= " port-number " certificate= "
certificate-name " loadBalancerProbe= "load-balancer-probe-name" idleTimeout
InMinutes= " tcp-timeout " />
    </Endpoints>
  </WorkerRole>

```

(2) 进行公共 IP 的超时设置时, .cscfg 更改如下:

```

<NetworkConfiguration>
  <VirtualNetworkSite name= "VNet" />
  <AddressAssignments>
    <InstanceAddress roleName= "VMRolePersisted" >
      <PublicIPs>
        <PublicIP name= "public-ip-name" idleTimeoutInMinutes= "timeout-in
-minutes" />
      </PublicIPs>
    </InstanceAddress>
  </AddressAssignments>
</NetworkConfiguration>

```

## 9.2 什么是 Azure Load Balancer

Azure Load Balancer 是一个第 4 层 (TCP、UDP) 的负载均衡器, 并且该负载均衡器可以在健康的虚机中合理分配来自外部或内部请求流量, 从而保证服务的高可用性。另外, 现在 Azure 负载均衡的类型分为两种: 外部负载均衡 & 内部负载均衡。

### 9.2.1 Azure 负载均衡的部署模型

#### 9.2.1.1 Azure 经典部署模型 (ASM)

在 ASM 模式中 (如图 9.2-1), 在一个云服务中所有的虚机通过一个公网 IP 地址和一个完全限定域名 FQDN 并组合一个负载均衡器来对网络流量进行负载均衡。负载均衡器会把流量经由之前定义好的终结点分配到虚机上的服务端口中, 并且负载均衡终结点在公共 IP 地址与分配到云服务中虚拟机上的服务的本地端口之间存在一对多的关系。

#### 9.2.1.2 Azure Resource Manager 部署模型 (新门户: 门户预览)

在 ARM 部署模型中 (如图 9.2-2), 无需创建云服务就可以把独立的公共 IP 地址资源与负载均衡器进行关联。然后, 将专用或公共 IP 地址分配到虚拟机上的服务端口, 并且将网络服务接口添加到负载均衡器的后端 IP 地址池中, 之后负载均衡器就可以根据所创建的负载均衡规则发送负载均衡的网络流量给相应的虚机。另外, 对于流入到负载均衡的网络流量, 还可以设置负载均衡的入站 NAT 规则作为外网访问虚机的终结点。

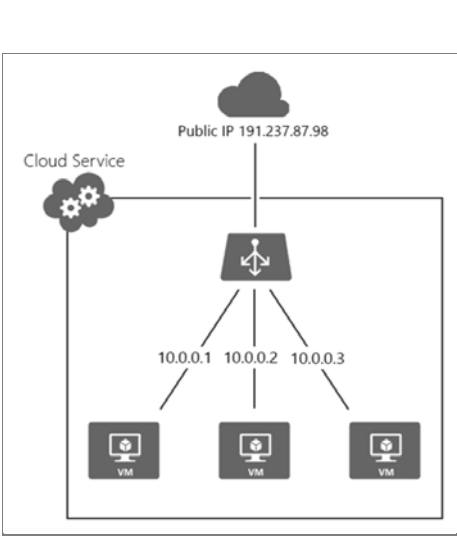


图 9.2-1

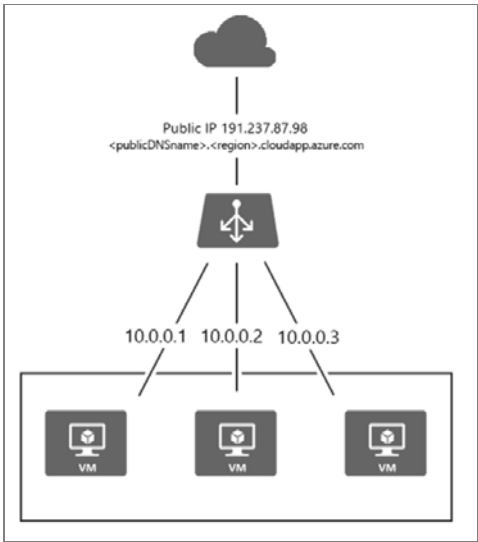


图 9.2-2

## 9.2.2 负载均衡器的分发模式

负载均衡器的分发模式分为三种：基于五元组哈希（源 IP、源端口、目标 IP、目标端口和协议类型），3 元组哈希（源 IP、目标 IP、协议）以及 2 元组哈希（源 IP、目标 IP）。

### 9.2.2.1 负载均衡器五元组哈希分发模式

一般配置负载均衡的默认分发模式为 5 元组（源 IP、源端口、目标 IP、目标端口和协议类型）哈希。这个分发模式（如图 9.2-3）会导致同一会话中的数据包会定向到经过负载均衡的终结点后面的同一个云服务或同一个订阅下的实例。另外，客户端从同一源 IP 重新发起会话时，源端口会发生更改，在这种情况下会导致流量重新定向到在同一负载均衡集中的其他虚机上。

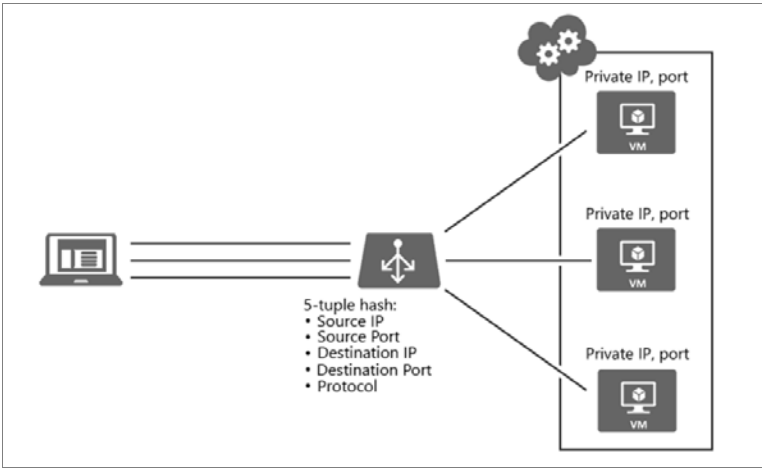


图 9.2-3



### 9.2.2.2 负载均衡器三元组或二元组哈希分发模式（基于客户端 IP 的会话关联）

负载均衡器三元组和二元组哈希分发模式（图 9.2-4）称为源 IP 会话关联（又称为客户端 IP），如果把负载均衡配置为使用三元组（源 IP、目标 IP）或二元组（源 IP、目标 IP、协议）来将流量映射到可用服务器的话，使用源 IP 关联会让同一客户端发起的连接请求重定向到同一台虚机上。

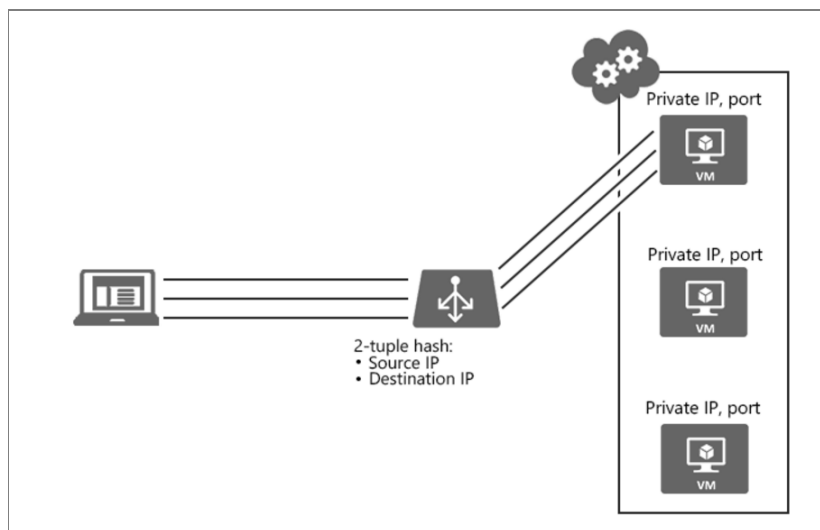


图 9.2-4

## 9.2.3 内部负载均衡

内部负载均衡与早前所介绍过的外部负载均衡有所不同。但是，不管是外部负载均衡，还是内部负载均衡，他们的原理机制都是基本相同的。不同点在于，外部负载均衡是针对外部进入到 Azure 虚机的流量进行负载分配，而内部负载均衡仅仅会被应用在 Azure 内网资源之间的数据通信（云服务或虚拟网络），或本地通过 VPN & ER 专线与 Azure 进行的内部数据交换。

内部负载均衡器（ILB）能够让内部业务在 Azure 中运行，并可以实现在 Azure 内部网络之间（同一个云服务内部之间或虚拟网络之间）或者本地与 Azure 之间互访相对应的应用程序，而这些应用程序绝对不会暴露在 Internet 中成为现实。

## 9.2.4 案例演示

### 9.2.4.1 面向内网 Intranet 的业务线应用程序部署架构

案例环境架构介绍（如图 9.2-5）：从本地客户端通过 P2S VPN 访问 ILB 后面的 Web 服务。另外，在该环境中除本地资源 Client（192.168.0.1）外，其余资源包括 Web Server-1（10.0.0.6），Web Server-2（10.0.0.7）以及 ILB（10.0.0.5）都部署在同一个 Azure 虚拟网络中。

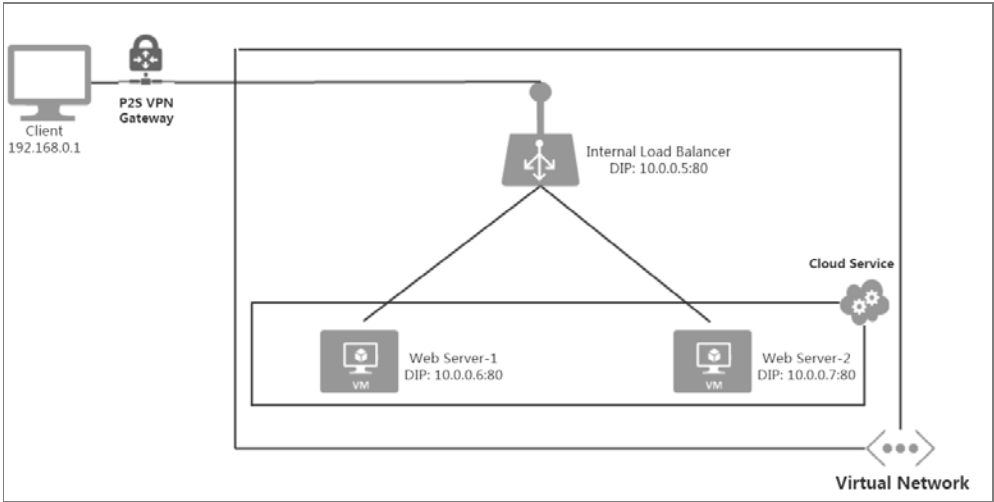


图 9.2-5

1. 具体操作步骤如下：

注：关于如何创建 VM，请参考第五章 5.1 节。关于如何搭建 P2S VPN 以及如何通过 P2S VPN 从本地登录到 Azure VM 的操作步骤，请参考第七章 7.3 节。另外，除创建 VM 以及搭建 P2S VPN 是通过 Portal 创建以后，其余资源全部是由 Azure PowerShell 来进行部署的。

创建 ILB 后端池中的两台 Web 服务器(Web Server-1 创建在虚机 winser08-3 上 & Web Server-2 创建在虚机 winser08-4 上)，如图 9.2-6 所示。

winser08-3	✓ 正在运行
winser08-4	✓ 正在运行

图 9.2-6

1) 在本地把 P2S VPN 配置好后进行 VPN 的连接，如图 9.2-7、图 9.2-8 和图 9.2-9 所示。

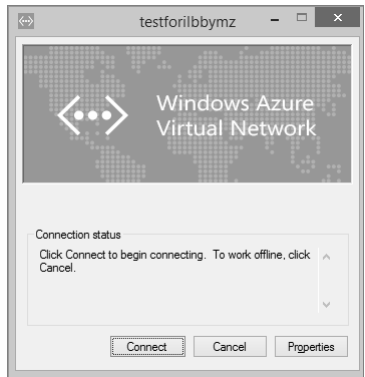


图 9.2-7

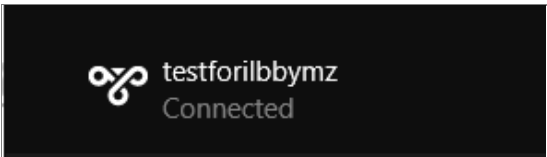


图 9.2-8

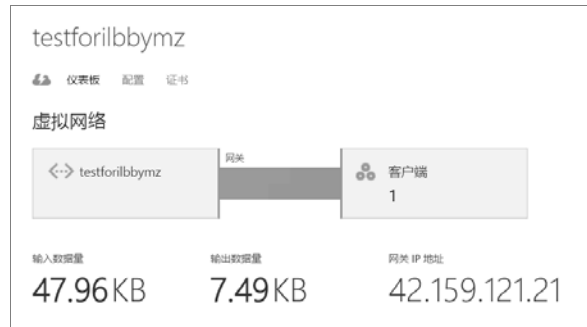


图 9.2-9

## 2) 在虚拟网络中创建 ILB 实例:

```
$svc= " testforilbbyzmz " ----- 云服务名称
$ilb= " ILB " -----为 ILB 实例命名
$subnet= " Subnet-1 " -----为 ILB 实例指定子网
$IP= " 10.0.0.5 " -----为 ILB 实例指定一个子网的 IP 地址，并且该地址不能被占用。
```

```
Add-AzureInternalLoadBalancer -ServiceName $svc -InternalLoadBalancerName
$ilb -SubnetName $subnet -StaticVNetIPAddress $IP
```

## 3) 查看 ILB 实例信息:

```
$svc= " testforilbbyzmz "
Get-AzureService -ServiceName $svc | Get-AzureInternalLoadBalancer
```

## 4) 为 Web Server-1 添加 ILB 的终结点:

```
$svc= " testforilbbyzmz "
$vmname= " winser08-3 " ----- 即将添加 ILB 终结点的虚拟机名称
$epname= " HTTPInternal " ----- ILB 终结点名称
$lbsetname= " ILBsetfor80 " ----- ILB 名称
$prot= " tcp " ----- 通信协议
$locport=80 ----- 私有端口
$pubport=80 ----- 公共端口
$ilb= " ILB "
```

```
Get-AzureVM -ServiceName $svc -Name $vmname | Add-AzureEndpoint -Name $epname
-Lbset $lbsetname -Protocol $prot -LocalPort $locport -PublicPort $pubport -
DefaultProbe -InternalLoadBalancerName $ilb | Update-AzureVM
```

## 5) 查看虚拟机是否加入到 ILB 中:

```
Get-AzureVM -ServiceName " testforilbbyzmz " -Name " winser08-3 " |
Get-AzureEndpoint
```

## 6) 为 Web Server-2 添加 ILB 的终结点:

```
$svc= " testforilbbyzmz "
$vmname= " winser08-4 "
```

```
$epname= " HTTPInternal "
$lbsetname= " ILBsetfor80 "
$prot= " tcp "
$locport=80
$pubport=80
$ilb= " ILB "
```

```
Get-AzureVM -ServiceName $svc -Name $vmname | Add-AzureEndpoint -Name $epname
-Lbset $lbsetname -Protocol $prot -LocalPort $locport -PublicPort $pubport -
DefaultProbe -InternalLoadBalancerName $ilb | Update-AzureVM
```

7) 查看虚拟机是否加入到 ILB 中:

```
Get-AzureVM -ServiceName " testforilbbymz " -Name " winser08-3 " |
Get-AzureEndpoint
```

8) 测试配置是否生效。

a) 证实可以通过 Client 访问 Web Server-1, 如图 9.2-10 所示。



图 9.2-10

b) 证实可以通过 Client 访问 Web Server-2, 如图 9.2-11 所示。



图 9.2-11

c) 每次通过 Client 访问 ILB 负载均衡的 80 端口会分配到不同的 IIS 服务器 (Web Server-1 或 Web Server-2) 上, 该现象可以证实 ILB 已生效, 如图 9.2-12 所示。

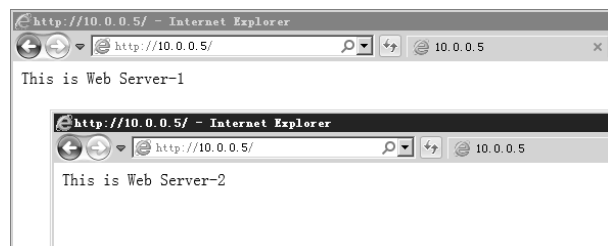


图 9.2-12

### 9.3 应用程序网关

Azure 应用程序网关是服务形式的应用程序传送控制器（ADC），借此为应用程序提供各种第 7 层负载均衡功能。它提供完全由 Azure 管理的高度可用、可缩放的服务。应用程序网关支持 SSL 卸载和端到端 SSL、基于 Cookie 的会话相关性、基于 URL 路径的路由、多站点托管，等等，如图 9.3-1 所示。

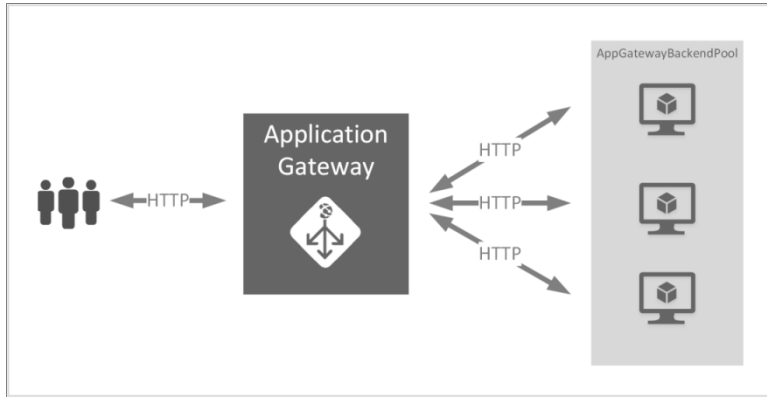


图 9.3-1

应用程序网关当前支持具有以下功能的第 7 层应用程序传送：

- **HTTP 负载均衡** - 应用程序网关提供轮循机制负载均衡。负载均衡在第 7 层完成，仅用于 HTTP (S) 流量。快速扩展你的大计算和大数据应用程序
- **基于 cookie 的会话相关性**：想要在同一后端保留用户会话时，此功能十分有用。借助受网关管理的 cookie，应用程序网关能够将来自用户会话的后续流量转到同一后端进行处理。
- **安全套接字层 (SSL) 卸载**：此功能让 Web 服务器免于执行解密 HTTPS 流量的高成本任务。通过在应用程序网关终止 SSL 连接，并将请求转发到未加密的服务器，Web 服务器不用承担解密的负担。应用程序网关会重新加密响应，然后再将它发回客户端。
- **端到端 SSL**：应用程序网关支持对流量进行端到端加密。应用程序网关通过在应用程序网关上终止 SSL 连接来完成此任务。网关随后将路由规则应用于流量、重新加密数据包，并根据定义的路由规则将数据包转发到适当的后端。来自 Web 服务器的任何响应都会经历相同的过程返回最终用户。
- **基于 URL 的内容路由**：此功能能够使用不同的后端服务器来处理不同的流量。可将 Web 服务器上的文件夹流量或 CDN 流量路由到不同的后端，让不提供特定内容的后端减少不必要的负载。
- **多站点路由**：应用程序网关允许在单个应用程序网关上合并最多 20 个网站。
- **WebSocket 支持**：应用程序网关的另一个重要功能是对 WebSocket 的本机支持。

- **运行状况监视** - 应用程序网关提供默认的后端资源运行状况监视，以及用于监视更多特定方案的自定义探测。

### 9.3.1 配置面向 Internet 的负载均衡

准备工作：

- 安装最新版本的 Azure PowerShell cmdlet。
- 使用现有或新建 VNET 下的一个空子网，专门供应用程序网关使用。
- 必须存在配置为使用应用程序网关的服务器，或者必须在虚拟网络中为其创建终结点，或者必须为其分配公共 IP/VIP。

应用程序网关组成部分：

- **后端服务器池**：后端服务器的 IP 地址列表。列出的 IP 地址应属于虚拟网络子网，或者是公共 IP/VIP。
- **后端服务器池设置**：每个池都有一些设置，例如端口、协议和基于 Cookie 的关联性。这些设置绑定到池，并会应用到池中的所有服务器。
- **前端端口**：此端口是应用程序网关上打开的公共端口。流量将抵达此端口，然后重定向到后端服务器之一。
- **侦听器**：侦听器具有前端端口、协议（Http 或 Https，这些值区分大小写）和 SSL 证书名称（如果要配置 SSL 卸载）。
- **规则**：规则将会绑定侦听器和后端服务器池，并定义当流量抵达特定侦听器时应定向到的后端服务器池。

面向 Internet 的负载均衡拓扑图，如图 9.3-2 所示。

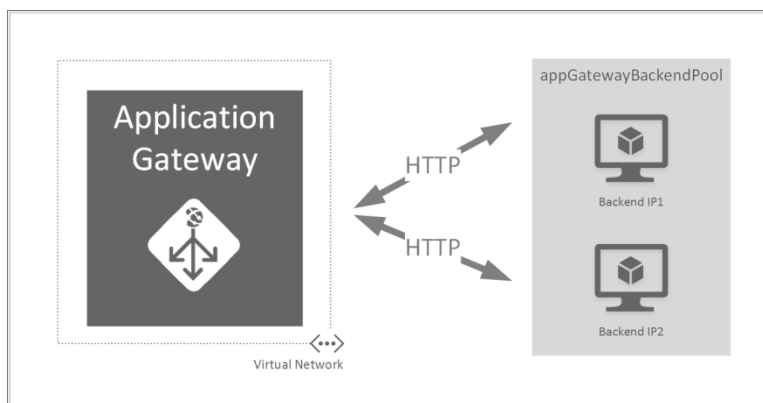


图 9.3-2

#### 9.3.1.1 创建应用程序网关

（1）创建网关，请使用 `New-AzureApplicationGateway` cmdlet，并将值替换为你自己的值。此时不会开始计收网关的费用。计费将在后面已成功启动网关时开始。

```
New-AzureApplicationGateway -Name XXXX -VnetName XXXX -Subnets @( " XXX " )
```

(2) 验证是否已创建网关，可以使用如下：

```
Get-AzureApplicationGateway XXXX
```

输出：

```
Name       : XXXX
Description :
VnetName   : testvnet1
Subnets    : {Subnet-1}
InstanceCount : 2
GatewaySize : Medium
State       : Stopped
VirtualIPs  : {}
DnsName     :
```

#### NOTE:

InstanceCount 的默认值为 2，最大值为 10。GatewaySize 的默认值为 Medium。你可以选择 Small、Medium 或 Large。

#### 9.3.1.2 配置应用程序网关

可以使用 XML 或配置对象配置应用程序网关，在以下示例中，使用 XML 文件配置应用程序网关设置，并将这些设置提交到应用程序网关资源，步骤如下：

(1) 将以下文本复制到记事本中，编辑配置项中括号之间的值，并使用扩展名为.xml 保存文件。

```
<?xml version="1.0" encoding="utf-8" ?>
<ApplicationGatewayConfiguration xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/windowsazure">
  <FrontendPorts>
    <FrontendPort>
      <Name>(name-of-your-frontend-port)</Name>
      <Port>(port number)</Port>
    </FrontendPort>
  </FrontendPorts>
  <BackendAddressPools>
    <BackendAddressPool>
      <Name>(name-of-your-backend-pool)</Name>
      <IPAddresses>
        <IPAddress>(your-IP-address-for-backend-pool)</IPAddress>
<IPAddress>(your-second-IP-address-for-backend-pool)</IPAddress>
      </IPAddresses>
    </BackendAddressPool>
  </BackendAddressPools>
  <BackendHttpSettingsList>
    <BackendHttpSettings>
      <Name>(backend-setting-name-to-configure-rule)</Name>
      <Port>80</Port>
```

```

        <Protocol>[Http|Https]</Protocol>
        <CookieBasedAffinity>Enabled</CookieBasedAffinity>
    </BackendHttpSettings>
</BackendHttpSettingsList>
<HttpListeners>
    <HttpListener>
        <Name>(name-of-the-listener)</Name>
        <FrontendPort>(name-of-your-frontend-port)</FrontendPort>
        <Protocol>[Http|Https]</Protocol>
    </HttpListener>
</HttpListeners>
<HttpLoadBalancingRules>
    <HttpLoadBalancingRule>
        <Name>(name-of-load-balancing-rule)</Name>
        <Type>basic</Type>

<BackendHttpSettings>(backend-setting-name-to-configure-rule)</BackendHttpSettings>

        <Listener>(name-of-the-listener)</Listener>

<BackendAddressPool>(name-of-your-backend-pool)</BackendAddressPool>
    </HttpLoadBalancingRule>
</HttpLoadBalancingRules>
</ApplicationGatewayConfiguration>

```

**NOTE:** 协议项 Http 或 Https 区分大小写。

(2) 设置应用程序网关。

将 Set-AzureApplicationGatewayConfig cmdlet 用于配置 XML 文件

```

#对应参数是应用程序网关的名字、xml 文件的路径
Set-AzureApplicationGatewayConfig -Name XXXX -ConfigFile "D:\XXX.xml "

```

(3) 启动应用程序网关。

```

网关设置好了之后，通过以下命令启动网关
Start-AzureApplicationGateway -Name XXXX

```

(4) 查看应用程序网关详细信息。

```

#获取网关详细信息，网关的公网 IP 已经生成
Get-AzureApplicationGateway -Name WinAppGW

```

(5) 验证应用程序网关。

此时应用程序网关已经配置完成，打开 IE 浏览器访问 DNS 域名或网关公网 IP，在刷新页面时可以发现对后端池轮询的负载均衡已经生效。

### 9.3.2 内部负载均衡

面向内部终结点的 Azure 应用程序网关，也称为内部负载均衡器（ILB）。配置使用 ILB 的网关适用于不向 Internet 公开的内部业务线应用程序。对于位于不向 Internet 公开的



安全边界内的多层应用程序中的服务和层也很有用，但仍需要执行轮循负载分布、会话粘性或安全套接字层（SSL）终止。

以下是创建应用程序网关所需的步骤：

- 创建新的应用程序网关。
- 配置网关。
- 设置网关配置。
- 启动网关。
- 验证网关。

### 9.3.2.1 创建新的应用程序网关

#### 1. 创建网关

```
New-AzureApplicationGateway -Name XXXX -VnetName XXX -Subnets XX
```

使用 `New-AzureApplicationGateway` cmdlet，并将值替换为你自己的值。请注意，此时不会开始计收网关的费用。计费将在后面已成功启动网关时开始。

#### 2. 验证网关

`Get-AzureApplicationGateway XXXX`

```
VERBOSE: 5:20:39 AM - Begin Operation:
Get-AzureApplicationGateway VERBOSE: 5:20:40 PM - Completed
Operation: Get-AzureApplicationGateway
Name: XXXX
Description:
VnetName: XXX
Subnets: XX
InstanceCount: 2
GatewaySize: Medium
State: Stopped
VirtualIPs:
DnsName:
```

### 9.3.2.2 配置网关

应用程序网关配置由多个值组成，这些值可将绑定在一起以构造配置。有效值如下：

- 后端服务器池：后端服务器的 IP 地址列表。列出的 IP 地址应属于 VNet 子网，或者是公共 IP/VIP。
- 后端服务器池设置：每个池具有端口、协议和基于 Cookie 的相关性等设置。这些设置绑定到池，并会应用到池中的所有服务器。
- 前端端口：此端口是应用程序网关上打开的公共端口。流量将抵达此端口，然后重定向到后端服务器之一。
- 侦听器：侦听器具有前端端口、协议（Http 或 Https，区分大小写）和 SSL 证书名称（如果要配置 SSL 卸载）。

- 规则：规则将会绑定侦听器 and 后端服务器池，并定义当流量抵达特定侦听器时应定向到的后端服务器池。目前仅支持基本规则。基本规则是一种轮循负载分发模式。通过 XML 文件构建配置

```
<?xml version="1.0" encoding="utf-8" ?>
  <ApplicationGatewayConfiguration>
.....
    <FrontendIPConfigurations>
      <FrontendIPConfiguration>
        <Name>fip1</Name>
        <Type>Private</Type>
        <StaticIPAddress>10.0.0.10</StaticIPAddress>
        .....
        <Name>FrontendPort1</Name>
        <Port>80</Port>
        .....
      <BackendAddressPool>
        <Name>BackendPool1</Name>
        <IPAddresses>
          <IPAddress>10.0.0.1</IPAddress>
          <IPAddress>10.0.0.2</IPAddress>
          .....
          <Port>80</Port>
          <Protocol>Http</Protocol>
          <CookieBasedAffinity>Enabled</CookieBasedAffinity>
.....
    <HttpListener>
      <Name>HTTPListener1</Name>
      <FrontendIP>fip1</FrontendIP>
      <FrontendPort>FrontendPort1</FrontendPort>
      <Protocol>Http</Protocol>
      .....
```

### 9.3.2.3 设置网关配置

使用上述的 XML 文件设置应用程序网关：

```
Set-AzureApplicationGatewayConfig -Name XXXX -ConfigFile D:\config.xml
```

### 9.3.2.4 启动网关

使用 Start-AzureApplicationGateway cmdlet 来启动网关：

```
Start-AzureApplicationGateway XXXX
```

输出结果：

```
VERBOSE: 5:39:16 AM - Begin Operation: Start-AzureApplicationGateway
VERBOSE: 5:39:52 AM - Completed Operation: Start-AzureApplicationGateway
Name      HTTP Status Code    Operation ID          Error
```

-----  
Successful OK

-----  
fc592db8-4c58-2c8e-9a1d-1c97880f0b9b

#### NOTE:

成功启动网关后，将开始计收应用程序网关的费用。

#### 9.3.2.5 验证网关

使用 `Get-AzureApplicationGateway` cmdlet 检查网关的状态：

Get-AzureApplicationGateway XXXX

如果前一步骤中的 `Start-AzureApplicationGateway` 成功，则 `State` 应为 `Running`，`Vip` 和 `DnsName` 应包含有效的条目。此示例在第一行显示 cmdlet，接着显示输出。在此示例中，网关正在运行并准备好接收流量，如下输出结果：

```
VERBOSE: 7:05:15 PM - Begin Operation: Get-AzureApplicationGateway
VERBOSE: 7:05:18 PM - Completed Operation: Get-AzureApplicationGateway
Name           : XXXX
Description    :
VnetName       : XXX
Subnets       : XX
InstanceCount  : 2
GatewaySize    : Medium
State          : Running
VirtualIPs     : {10.0.0.10}
DnsName        :
lytappgw-b7a11563-2b3a-5172-b4aa-206ee5c23edd.chinacloudapp.cn
```

#### NOTE:

在此示例中，应用程序网关配置为在配置的 ILB 终结点 10.0.0.10 上接受流量。

### 9.3.3 SSL OffLoad

Application Gateway 起到了 SSL 加解密的作用，客户端跟 App Gateway 之间 SSL Session 交互，不需要跟后台的所有的 Web 服务器分别建立 SSL session，所有的 SSL 行为和 SSL 证书统一在 App Gateway 设备上统一管理维护，相当于为后端 VM 卸载掉 SSL 加密的任务量，释放了后端 VM 的消耗在 SSL 加密上的资源。如图 9.3-3 所示的拓扑图。

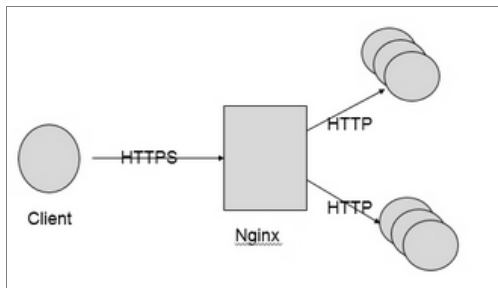


图 9.3-3

**准备工作：**

若要配置应用程序网关的 SSL 卸载，必须提供证书。此证书在应用程序网关上加载，用于加密和解密通过 SSL 发送的流量。证书需采用个人信息交换（pfx）格式。此文件格式适用于导出私钥，后者是应用程序网关对流量进行加解密所必需的。

**9.3.3.1 配置 SSL 卸载**

1. 上传证书到应用程序网关（制定网关名称、证书名称、密码、证书路径）：

```
Add-AzureApplicationGatewaySslCertificate -Name XXXX -CertificateName XXX
-Passsword XXX -CertificateFile D:\httpscert.pfx
```

2. 配置 XML 文件，参考如下：

**(1) FrontendPorts:**

用来定义应用程序网关上的公网端口，我们要测试 SSL Offload，所以定义 FrontendPort1 为 443 端口：

```
<FrontendPorts>
<FrontendPort>
<Name>FrontendPort1</Name>
<Port>443</Port>
</FrontendPort>
</FrontendPorts>
```

**(2) BackendAddressPools:**

用来定义后端 Web 服务器集群的地址群，我们定义了 BackendPool1，它包含 10.0.0.4 和 10.0.1.4 两个 AzureVM:

```
<BackendAddressPool>
<Name>BackendPool1</Name>
<IPAddresses>
<IPAddress>10.0.0.4</IPAddress>
<IPAddress>10.0.1.4</IPAddress>
</IPAddresses>
</BackendAddressPool>
```

**(3) BackendHttpSettingsList:**

用来定义端口、协议、cookie-based affinity，此处我们定义了 BackendSetting1，它包含了 Web 服务器 80 端口、使用 http 协议、禁用 cookie-based affinity:

```
<BackendHttpSettingsList>
<BackendHttpSettings>
<Name>BackendSetting1</Name>
<Port>80</Port>
<Protocol>Http</Protocol>
<CookieBasedAffinity>Disabled</CookieBasedAffinity>
</BackendHttpSettings>
</BackendHttpSettingsList>
```

#### (4) HttpListeners:

用来定义监听器，它起到监听应用程序网关公网端口的作用，用来响应公网用户的请求，此处我们定义了 HTTPListener1，监听 FrontendPort1（443），使用 Https 协议、证书名字是 GWCert。

```
<HttpListeners>
<HttpListener>
<Name>HTTPListener1</Name>
<FrontendPort>FrontendPort1</FrontendPort>
<Protocol>Https</Protocol>
<SslCert>GWCert</SslCert>
</HttpListener>
</HttpListeners>
```

#### (5) HttpLoadBalancingRules:

用来定义负载均衡规则，此处我们定义一个规则名为 HttpLBRule1，它使用 basic 规则（轮询的负载分配机制），绑定 BackendSetting1（Web 服务器 80 端口、使用 http 协议、禁用 cookie-based affinity），绑定 HTTPListener1（监听 FrontendPort1（443），使用 Https 协议、证书名字是 GWCert），绑定 BackendPool1（包含 10.0.0.4 和 10.0.1.4 两个 AzureVM），所以 HttpLoadBalancingRules 可以说是所有信息的汇总，让我们完成 SSL Offload+负载均衡。

```
<HttpLoadBalancingRules>
<HttpLoadBalancingRule>
<Name>HttpLBRule1</Name>
<Type>basic</Type>
<BackendHttpSettings>BackendSetting1</BackendHttpSettings>
<Listener>HTTPListener1</Listener>
<BackendAddressPool>BackendPool1</BackendAddressPool>
</HttpLoadBalancingRule>
</HttpLoadBalancingRules>
</ApplicationGatewayConfiguration>
```

### 3. 配置应用程序网关:

通过修改好的 xml 配置应用程序网关，使我们的证书生效:

```
Set-AzureApplicationGatewayConfig -Name XXXX -ConfigFile D:\ssloffload.xml
```

输出结果:

```
PS C:\Users\luyitong> Set-AzureApplicationGatewayConfig -Name XXXX
-ConfigFile D:\ssloffload.xml
Name HTTP Status Code Operation ID Error
----
Successful OK df592db8-4c58-2c8e-9a1d-1c97880f022b
```

### 9.3.4 基于 URL 的访问设置

基于 URL 路径的路由，可根据 Http 请求的 URL 路径来关联路由。它将检查是否有路由由连接到针对应用程序网关中的 URL 列表配置的后端池，并将网络流量发送到定义的后端池。基于 URL 的路由的常见用法是将不同内容类型的请求负载均衡到不同的后端服务器池。

在以下示例中，应用程序网关针对 contoso.com 从三个后端服务器池提供流量，例如：VideoServerPool、ImageServerPool 和 DefaultServerPool，如图 9.3-4 所示。

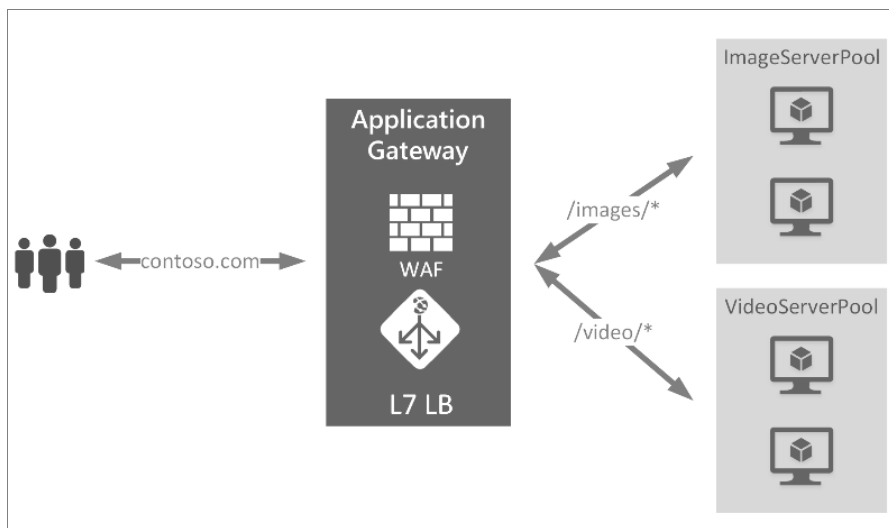


图 9.3-4

对 `http://contoso.com/video*` 请求将路由到 VideoServerPool，对 `http://contoso.com/images*` 的请求将路由到 ImageServerPool。如果没有任何路径模式匹配则选择 DefaultServerPool。

#### 9.3.4.1 UrlPathMap 配置元素

UrlPathMap 元素是用于指定后端服务器池映射的路径模式。这是模板文件中 urlPathMap 元素的代码段。

```
"urlPathMaps": [
{
  "name": "<urlPathMapName>",
  "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/urlPathMaps/<urlPathMapName>",
  "properties": {
    "defaultBackendAddressPool": {
      "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/backendAddressPools/<poolName>"
    },
    "defaultBackendHttpSettings": {
      "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/backendHttpSettingsList/<settingsName>"
    }
  }
}
```

```

    },
    "pathRules": [
      {
        "paths": [
          <pathPattern>
        ],
        "backendAddressPool": {
          "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/backendAddressPools/<poolName2>"
        },
        "backendHttpsettings": {
          "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/backendHttpsettingsList/<settingsName2>"
        }, .....
      }
    ]
  }
}

```

**NOTE:**

**PathPattern:** 这是要匹配的路径模式列表。每个模式必须以 / 开始，只允许在后接“/”的末尾处添加\*。发送到路径匹配器的字符串不会在第一个 ? 或 # 之后包含任何文本，这些字符在这里是不允许的。

## 9.3.4.2 PathBasedRouting 规则

PathBasedRouting 类型的 RequestRoutingRule 可用于将侦听器绑定到 urlPathMap。针对此侦听器收到的所有请求将根据 urlPathMap 中指定的策略进行路由。PathBasedRouting 规则的代码段：

```

"requestRoutingRules": [
  {
    "name": "<ruleName>",
    "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/requestRoutingRules/<ruleName>",
    "properties": {
      "ruleType": "PathBasedRouting",
      "httpListener": {
        "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/httpListeners/<listenerName>"
      },
      "urlPathMap": {
        "id": "/subscriptions/<subscriptionId>/../microsoft.network/applicationGateways/<gatewayName>/urlPathMaps/<urlPathMapName>"
      }, .....
    }
  }
]

```

## 9.3.5 案例分析

## 9.3.5.1 问题描述

用户部署了由四个实例组成的应用程序网关，前端开放 443 端口并通过应用程序网关做 SSL 卸载，后端池中的两台服务器部署了 IIS 服务并设置 8443 作为监听端口。当用户通

过应用程序网关访问网页时会报 502 的错误,但是直接访问后端 IIS 服务器就不会出现问题。

具体的部署架构以及各个有关服务器的 IP 地址请见如图 9.3-5 所示。

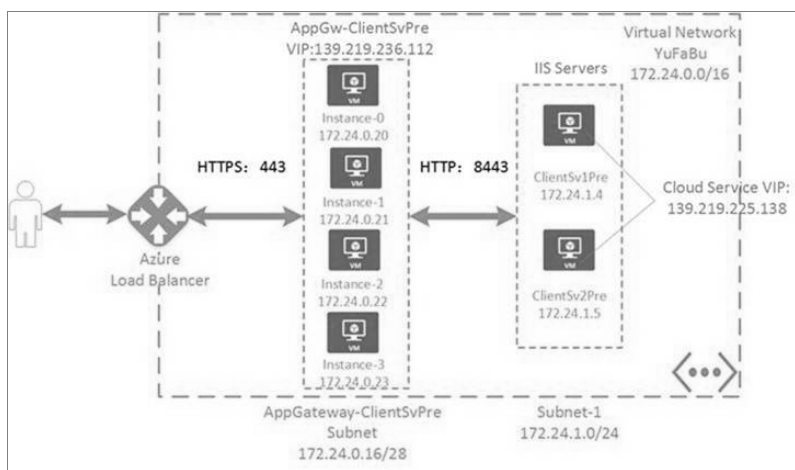


图 9.3-5

### 9.3.5.2 问题分析

IIS 服务器进行了域名绑定,监听 8443 端口绑定了域名 pre-paytest.bj.com,如图 9.3-6 所示。



图 9.3-6

同时,在应用程序网关配置了自定义探测,Host 不管是指定为 pre-paytest.bj.com 还是 pre-paytest.bj.com:8443 都无法解决此问题。

```
'Probes': [{ 'Name': 'Probe01', 'Protocol': 0, 'Host': 'pre-paytest.bj.com:8443', 'Path': '/', 'Interval': 15, 'Timeout': 15, 'UnhealthyThreshold': 5 },
  { 'Name': 'BackendSetting7', 'Port': 8443, 'Protocol': 0, 'CookieBasedAffinity': 'Enabled', 'RequestTimeout': 1800, 'Probe': 'Probe01' },
```

通过在后端 IIS 服务器中抓取的网路数据包可以发现,后端 IIS 服务器对应用程序网关发送的探测包回复了 500 的 HTTP 包,这是应用程序网关返回 502 报错的直接原因。

```
Frame: Number = 1447, Captured Frame Length = 145, MediaType = ETHERNET
+ Ethernet: Etype = Internet IP (IPv4), DestinationAddress: [00-17-FA-00-5E-DF], SourceAddress: [00-1C-73-98-C8-29]
```



```

+ Ipv4: Src = 172.24.0.20, Dest = 172.24.1.4, Next Protocol = TCP, Packet
ID = 22388, Total IP Length = 131
+ Tcp:  Flags=...AP..., SrcPort=49236, DstPort=8443, PayloadLen=91,
Seq=3832417169 - 3832417260, Ack=3877178317, Win=4121 (scale factor 0x8) =
1054976
- Http: Request, GET /
  Command: GET
+ URI: /
  ProtocolVersion: HTTP/1.1
  Connection: Keep-Alive
  Host: pre-paytest.bj.com:8443
  Max-Forwards: 10
HeaderEnd: CRLF
Frame: Number = 1448, Captured Frame Length = 4348, MediaType = ETHERNET
+ Ethernet:  Etype   = Internet   IP   (IPv4), DestinationAddress:
[12-34-56-78-9A-BC], SourceAddress:[00-17-FA-00-5E-DF]
+ Ipv4: Src = 172.24.1.4, Dest = 172.24.0.20, Next Protocol = TCP, Packet
ID = 28163, Total IP Length = 0
+ Tcp:  Flags=...AP..., SrcPort=8443, DstPort=49236, PayloadLen=4294,
Seq=3877178317 - 3877182611, Ack=3832417260, Win=4121 (scale factor 0x8) =
1054976
- Http: Response, HTTP/1.1, Status: Internal server error, URL: /
  ProtocolVersion: HTTP/1.1
  StatusCode: 500, Internal server error
  Reason: Internal Server Error
  Cache-Control: private
+ ContentType: text/html; charset=utf-8
  Server: Microsoft-IIS/8.5
  XAspNetVersion: 4.0.30319
  XPoweredBy: ASP.NET
  Date: Thu, 09 Mar 2017 10:42:53 GMT
  ContentLength: 4054
  HeaderEnd: CRLF
+ payload: HttpContentType = text/html; charset=utf-8

```

而当我们通过内网中的另外一台虚机（非应用程序网关后端池中的虚机）对该 IIS 服务器进行访问时，却可以实现正常访问。查看 HTTP 请求的包头，唯一的区别就是客户端直接访问的 HTTP 请求比应用程序网关发出的请求多了用户信息，如图 9.3-7 所示。

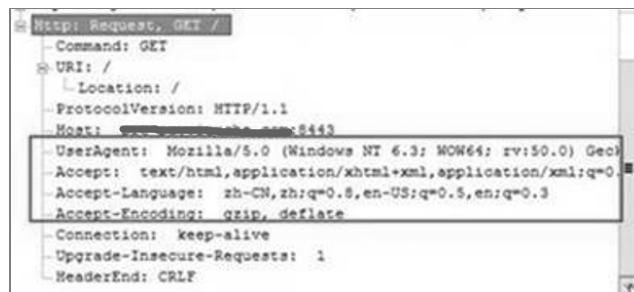


图 9.3-7

同时从 IIS 服务器环境中得到的另外一个测试结果，可以发现如果真的是由于请求的域名不正确的话，那么服务器应该返回 404 的报错，而不是 500 的报错。

鉴于以上分析，这个问题是 IIS 对于携带主机信息的应用程序请求访问包处理异常所导致的这个问题。所以，需要继续从 IIS 层面来继续排查这个问题。

通过分析 IIS Log，发现问题主要是由于用户的代码 **bj.Cashier.ApiClient.MultiLang RouteHandler.GetHttpHandler** 返回了 **NullReferenceException** 的异常而导致了这个问题。

该错误的具体解析如下（如图 9.3-8 所示）：



图 9.3-8

回过头来，从之前的分析中总结出发生 Error 500 的 HTTP 请求包（应用程序网关所发出 HTTP 请求包）信息如下显示：

```
Frame: Number = 1447, Captured Frame Length = 145, MediaType = ETHERNET
+ Ethernet: Etype = Internet IP (IPv4), DestinationAddress:
[00-17-FA-00-5E-DF], SourceAddress:[00-1C-73-98-C8-29]
+ Ipv4: Src = 172.24.0.20, Dest = 172.24.1.4, Next Protocol = TCP, Packet
ID = 22388, Total IP Length = 131
+ Tcp: Flags=...AP..., SrcPort=49236, DstPort=8443, PayloadLen=91,
Seq=3832417169 - 3832417260, Ack=3877178317, Win=4121 (scale factor 0x8) =
1054976
- Http: Request, GET /
  Command: GET
+ URI: /
  ProtocolVersion: HTTP/1.1
  Connection: Keep-Alive
  Host: pre-paytest.bj.com:8443
  Max-Forwards: 10
HeaderEnd: CRLF
```

而正常工作的 HTTP 请求包如图 9.3-9 所示，多了一些用户的信息。

所以，通过分析用户这边提供的代码，最终得到这个问题很可能出现在 **lang = request Context.HttpContext.Request.UserLanguages[0].ToString();** 这一段代码上。

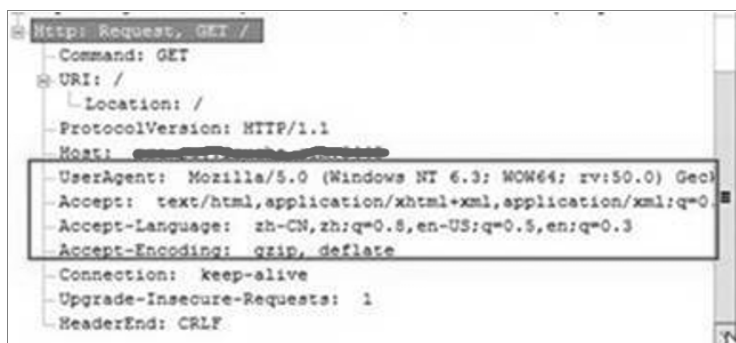


图 9.3-9

由于应用程序网关发送的探测包并不包含 Cookie 信息也不包含 User Languages 信息，所以在执行下段代码时会抛出空指针（**NullReferenceException**）的异常。

```

        if (requestContext.HttpContext.Request.Cookies[ " lang " ] !=
null)
        {
            lang = requestContext.HttpContext.Request.Cookies[ " lang " ].
Value.ToString();
        }
        else
        {
            lang = requestContext.HttpContext.Request.UserLanguages[0].
ToString();
        }

```

另外，通过 WFetch 工具做了一些测试也可以证明这个问题跟 UserLanguages 是有关系的。具体的测试结果如下：

（1）后端池服务器开放了 8443 端口在公网 IP 139.219.225.138 上，并且 Host 文件加上域名绑定之后绕过应用程序网关来测试 Web 站点，如图 9.3-10。

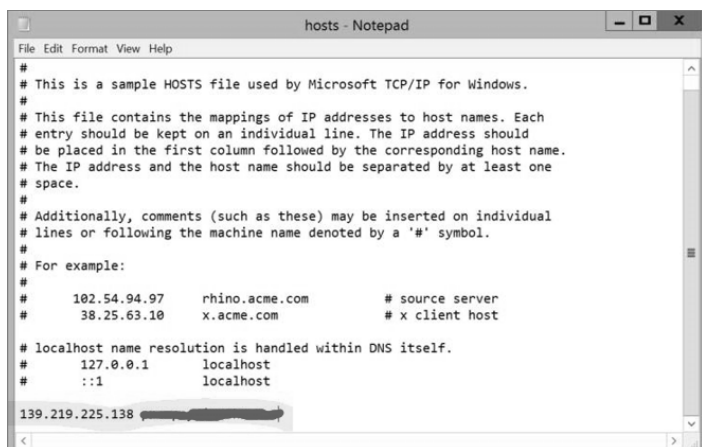


图 9.3-10

在 WFetch 不加自定义 Header 访问网页会报 500 的错误，如图 9.3-11 所示。



图 9.3-11

不加定义的 Http Header 信息如下，其中不包含 UserLanguages 信息：

```
Frame: Number = 627, Captured Frame Length = 111, MediaType = ETHERNET
+ Ethernet: Etype = Internet IP (IPv4), DestinationAddress: [2E-21-72-B8-DB-61],
SourceAddress: [00-1D-D8-B7-58-1B]
+ Ipv4: Src = 117.185.8.102, Dest = 139.219.225.138, Next Protocol = TCP,
Packet ID = 32110, Total IP Length = 97
+ Tcp: Flags=...AP..., SrcPort=59292, DstPort=8443, PayloadLen=57,
Seq=1647107330 - 1647107387, Ack=1604747273, Win=4134
- Http: Request, GET /
  Command: GET
+ URI: /
  ProtocolVersion: HTTP/1.1
  Host: pre-paytest.bj.com:8443
  Accept: */*
  HeaderEnd: CRLF
```

(2) 但是，在加上 Accept-Language: en-US\r\n 后进行访问时，就不会再次报错了，如图 9.3-12 所示。

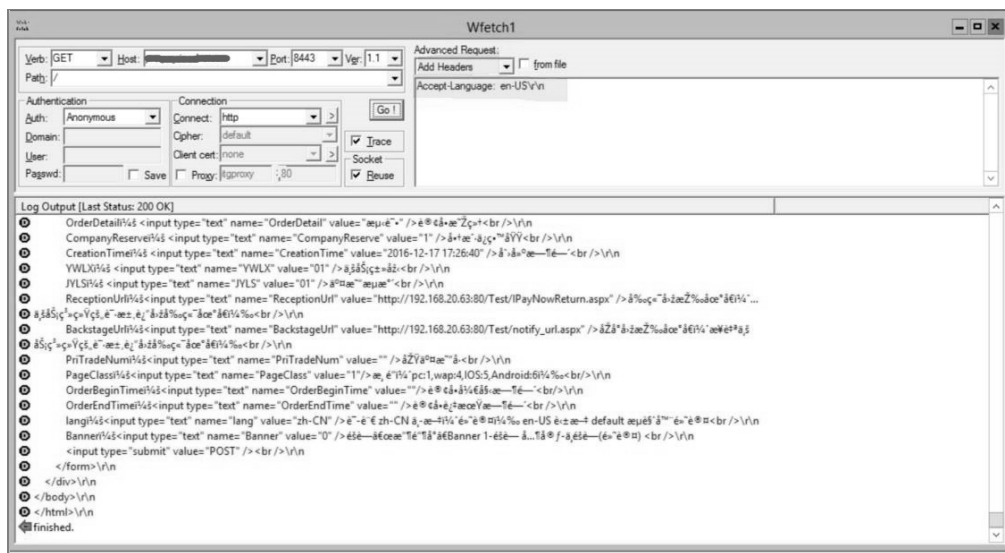


图 9.3-12

而且, 从抓包来看, 如下 Http header 上包含 UserLanguages 信息 (Accept-Language: en-US):

```

Frame: Number = 442, Captured Frame Length = 136, MediaType = ETHERNET
+ Ethernet: Etype = Internet IP (IPv4), DestinationAddress: [2E-21-72-B8-DB-61],
SourceAddress: [00-1D-D8-B7-58-1B]
+ Ipv4: Src = 117.185.8.102, Dest = 139.219.225.138, Next Protocol = TCP,
Packet ID = 32119, Total IP Length = 122
+ Tcp: Flags=...AP..., SrcPort=59319, DstPort=8443, PayloadLen=82,
Seq=1094165775 - 1094165857, Ack=2470797914, Win=4140
- Http: Request, GET /
  Command: GET
+ URI: /
  ProtocolVersion: HTTP/1.1
  Accept-Language: en-US
  Host: pre-paytest.bj.com:8443
  Accept: */*
HeaderEnd: CRLF

```

### 9.3.5.3 问题原因和解决方案

问题主要是因为用户的代码 `bj.Cashier.ApiClient.MultiLang.RouteHandler.GetHttpHandler` 针对应用程序网关对后端地址池 IIS 服务器的访问返回了 `NullReferenceException` 的异常而导致了这个问题。

经过排查得知用户的代码会进行如下逻辑判断:

- (1) 如果 Cookie 字段的 Language 不为空, 则使用 Cookie 字段的 Language;
- (2) 如果 Http header 字段的 Language 不为空, 则使用 Http header 字段的 Language;
- (3) 如果 Http header 字段的 language 为空, 则返回 `NullReferenceException`;

针对以上逻辑，如果后端 IIS Server 收到的 Http 请求即无 Cookie，而且 Http header 中又没有 Language 字段，则后端服务器的 Web 应用就会响应异常。而应用程序网关向后端地址池发出访问的报文中恰好即没有 Cookie，而且 Http header 中也不包含 language 字段，故后端地址池针对应用程序网关的访问不能正常的响应 Http 200，所以导致通过该应用程序网关无法正常访问 Web 服务。

对于这个问题，建议用户在代码中加入对应的逻辑判断，即如果 Cookie 字段为空，又加之如果 Http header 字段的 language 为空的话，则 Web 应用使用默认的 Language 响应 Http 查询请求。然后这个问题就可以得到解决。

## 9.4 可用性集

### 9.4.1 什么是可用性集

创建在可用性集中的 VM 将在底层基础结构中把不同的 VM 放置在不同的物理设备，这样做的意义在于如果 Azure 平台发生计划内维护事件或者计划外底层硬件发生故障的话，至少可以保证一个 VM 处于运行的状态，如图 9.4-1 所示。

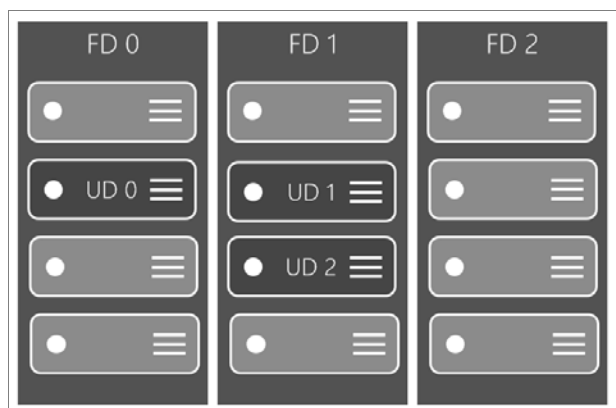


图 9.4-1

可用性集中的每个虚拟机会被分配一个更新域和一个容错域。

更新域（UD: Updated Domain）定义为：在默认情况下会分配 5 个非用户可配置的更新域（Resource Manager 可以部署最多 20 个更新域），这里指的是可同时重启的虚拟机和基础物理硬件组。如果单个可用性集中配置了超过 5 个虚拟机，第 6 个虚拟机将放置在第 1 个虚拟机所在的更新域中，第 7 个虚拟机将放置在第 2 个虚拟机所在的更新域中，依此类推。在计划内维护期间，更新域的重启顺序可能不会按照顺序进行，但一次只重启一个更新域中的虚拟机。

容错域（FD: Fabric Domain）定义为：一组共用一个通用电源和网络交换机的虚拟机。默认情况下，在可用性集中配置的虚拟机隔离在最多三个容错域（经典部署为两个容错域）中。虽然将虚拟机置于可用性集中并不能让你的应用程序免受由于操作系统或应用程序的故障所造成的影响，但可以限制潜在物理硬件故障、网络中断或电源中断的影响。

### 9.4.2 什么是计划内与计划外维护

当 Azure 平台发生计划内或计划外的事件时，只包含单个 VM 的可用性集是得不到任何保护的。Azure SLA 要求是一个可用性集中至少有两个或两个以上的 VM 才能实现至少有一台 VM 是可用的状态。

现在有两种事件可能影响到虚拟机的可用性：计划内维护和计划外维护。

(1) 计划内维护事件是指对平台进行的定期维护更新，以改进平台总体的可靠性、性能以及安全性。大多数情况下这些更新是不会影响到虚拟机或云服务。但有时，这些更新要求虚拟机进行重启。

(2) 计划外维护事件是关于虚机所在物理架构上所发生的一系列的故障情况。检测到此类故障时，Azure 平台会自动将虚拟机从所在的不正常物理机上迁移到正常的物理机上。如发生此类事件的话，会导致虚拟机的重启。

设计服务架构时（如图 9.4-2），请将同一目的前端池服务（比如：IIS）的 VM 放在同一个可用集中。然后，将同一目的的后端池服务（比如：SQL 数据库）的 VM 放在另外一个可用性集中。这样做的目的在于，如果 VM 在遇到计划内或计划外的维护事件的时候，可以保证前端池和后端池的 VM 至少有一台是可用的。

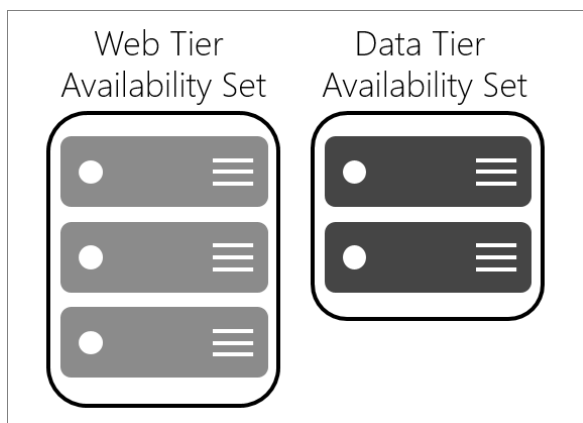


图 9.4-2

在对于存储高可用性的设计来说。最佳做法是对可用性集中的每个 VM 使用单独的存储账户。同一可用性集中的多个 VM 绝对不能共享存储账户，但不同可用性集之间的 VM 可共享存储账户。总之在 Azure 存储账户中，每个磁盘（VHD）是一个页 blob。请务必确保存储账户之间存在冗余和隔离，以便为可用性集中的 VM 提供高可用性。

另外，建议您使用负载均衡器配合可用性集来一起，这样可以保证不管发生计划内或计划外维护事件，始终会有至少一台 VM 是可用的。

### 9.4.3 如何在 ASM 模式下为在同一个云服务下的两台虚机去创建可用性集

(1) 首先，创建两台虚机。如图 9.4-3 所示。



图 9.4-3

（2）选择虚拟机 winser08-3，然后单击“配置”，并选择创建可用性集。然后为可用性集命名后，通过单击“保存”来创建可用性集。如图 9.4-4 和图 9.4-5 所示。



图 9.4-4



图 9.4-5



(3) 把 winser08-4 添加到之前已创建好的可用性集中，并保存该设置。如图 9.4-6 所示。



图 9.4-6

(4) 现在可以看到两台虚机（winser08-3 & winser08-4）都已添加到可用性集中。而且，您还可以在这两台虚机所属的那个云服务中去单击“实例”来查看这两台虚机所属更新域和容错域的位置。如图 9.4-7 和图 9.4-8 所示。



图 9.4-7



图 9.4-8

## 9.5 Autoscale\VMSS

虚拟机规模集是一种 Azure 计算资源，可用于部署和管理一组相同的 VM。VM 规模集中的所有 VM 均采用相同的配置，而无需对 VM 进行预配，这可更简便地生成面向大计算、大数据、容器化工作负荷的大规模服务。

对于需要扩大和缩小计算资源的应用程序，缩放操作在容错域和更新域之间进行隐式平衡。虚拟机规模集可让你以集的形式管理多个 VM。概括而言，规模集具有以下特点：

- (1) 几分钟内创建上百个相同的虚拟机。
- (2) 快速扩展你的大计算和大数据应用程序。
- (3) 依赖于集成负载均衡和自动扩展。
- (4) 高可用性。每个规模集将它的 VM 放入具有 5 个容错域(FD)和 5 个更新域(UD)的可用性集，以确保可用性。
- (5) 简化 VM 的部署、管理和清理。
- (6) 支持常用的 Windows 和 Linux 版本以及自定义映像。

### 9.5.1 创建和管理 VM 规模集

可以在 Azure 门户预览中选择“新建”，然后在搜索栏中键入“规模”，来创建 VM 规模集。结果中会看到“虚拟机规模集”。从这里，可以填写必填字段，自定义和部署规模集。请注意，门户中也提供了基于 CPU 使用情况设置自动调整规模的基本规则的选项。

也可以使用 JSON 模板和 REST API 定义和部署 VM 规模集，就像定义和部署单个 Azure Resource Manager VM 一样。因此，可以使用任何标准的 Azure 资源管理器部署方法。

- (1) 首先，在 Web 浏览器中登录到 Azure 门户预览。在“新建”栏下方选择“虚拟机规模集”条目。
- (2) 使用默认设置并快速创建规模集。

- 1) 在基本边栏选项卡上，输入规模集名称。
- 2) 选择所需的 OS 类型，输入用户名、密码，选择使用的“订阅”。
- 3) 输入所需的资源组名称和位置，然后单击 OK，如图 9.5-1 所示。



图 9.5-1

- (3) 在 Virtual machine scale set service settings 边栏选项卡上：输入所需的域名标签（规模集前端负载均衡器的 FQDN 的基础）。在整个 Azure 中，此标签必须是唯一的。
- 选择所需的操作系统磁盘映像、实例计数和 VM 大小，如图 9.5-2 所示。

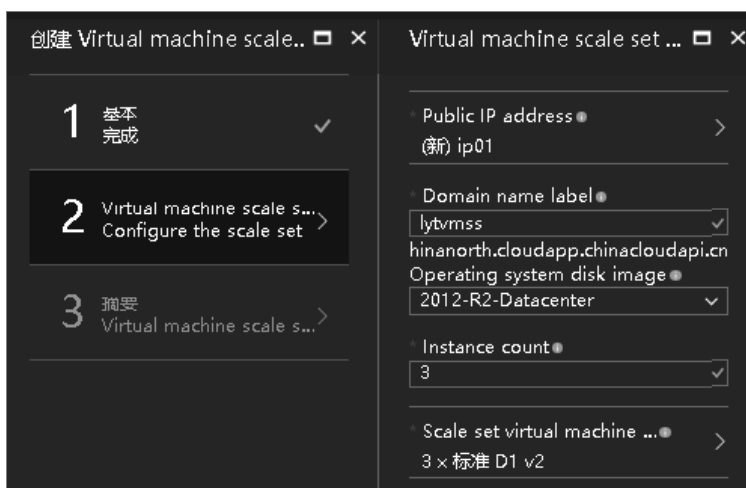


图 9.5-2

(4) 验证完成后，在 Summary 边栏选项卡上，单击 OK 开始进行规模集部署，如图 9.5-3 所示。



图 9.5-3

(5) 查看部署状态，如图 9.5-4 所示。

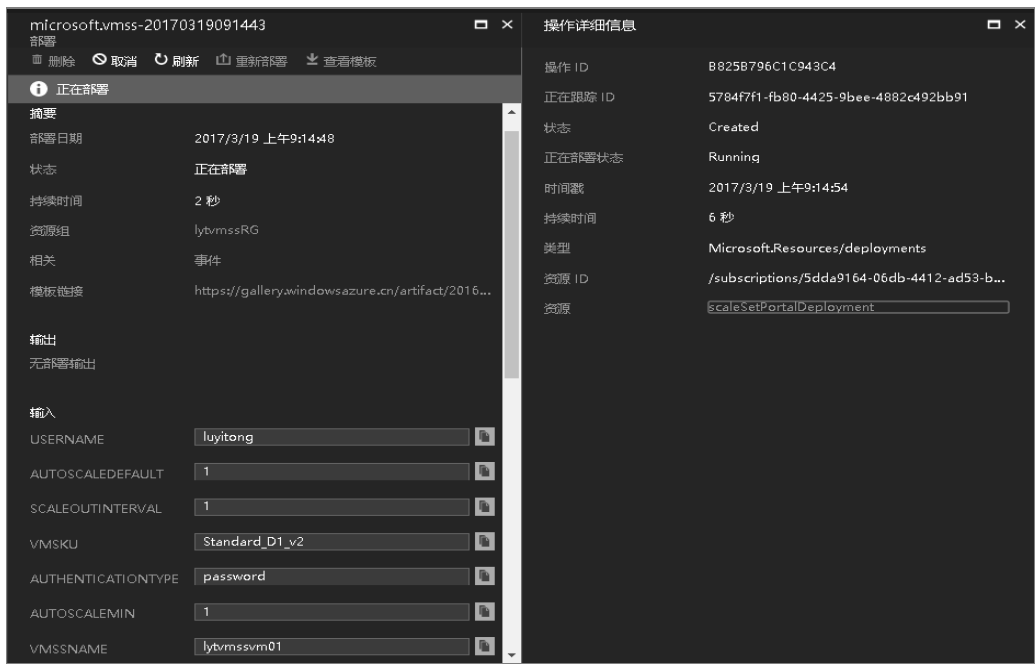


图 9.5-4

(6) 查看该规模集资源组下成功创建的全部资源，如图 9.5-5 所示。



图 9.5-5

**备注：**

- 通过 RDP/SSH 连接到 VM 规模集实例 - VM 规模集是在 VNET 中创建的，并且没有为规模集中单独的 VM 分配公共 IP 地址。这是一件好事，因为您通常不希望承担为计算网格中的所有无状态资源分配单独的公共 IP 地址而产生的支出和管理开销，并且您可以轻松地从 VNET 中的其他资源（包括负载均衡器或独立虚拟机等具有公共 IP 地址的资源）连接到这些 VM。
- 使用 NAT 规则连接到 VM - 可以创建一个公共 IP 地址，并将其分配给负载均衡器，然后定义入站 NAT 池，用于将 IP 地址上的端口映射到 VM 规模集中的 VM 上的端口。例如：

源	Source Port	目标	Destination Port
公共 IP	端口 50000	vm ss 0	端口 22
公共 IP	端口 50001	vm ss_1	端口 22
公共 IP	端口 50002	vm ss_2	端口 22

**9.5.2 案例一**

**现状：**C 公司是一家电商企业，用户的访问，峰值时间都是很难预测的，尤其是在重要的节假日或商业促销的时候，传统数据中心到底应该部署什么规模的 Web 集群一直是一个问题，如果部署过量会造成成本和资源的浪费，部署过少，则会在遇到峰值时来不及扩充，容易造成用户无法访问、用户体验差、交易额损失等等，同样，该公司的 IT 运维人员在这个时期就会面临神经紧绷、实时检测的压力情况……

**方案：**VMSS 作为 Azure 云服务新的计算方式，提供了根据服务压力负载自动扩展收缩，并且同时能够支持 Windows 和 Linux 系统，在提供了 IaaS 级别的控制灵活性的同时，也提供了 PaaS 级别的自动扩展，对于无状态的 Web 应用服务等场景非常适合。针对 C 公

司目前遇到的现状，该企业 IT Manager 决定将本地数据中心 Web 应用服务迁移到 Azure IaaS 平台，通过 ARM 模板和 VMSS 创建一个自动负载均衡的，按照 CPU 负载自动扩展的 Web 服务器集群，如图 9.5-6 所示。

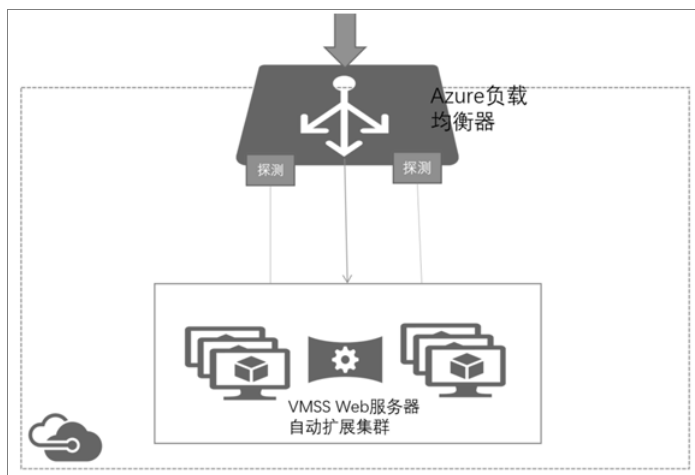


图 9.5-6

在本模板中，将会创建如下实验环境：

- 定义一个负载均衡器，负责转发前端的 Web 请求给后端的 Web 集群。
- 使用 VMSS 创建一个 Web 集群。
- 使用客户定制化脚本，自动安装 Apache Web 服务器，和 PHP Web 应用。
- 定义自动扩展集合的规则，根据虚拟机自动扩展集合中的 CPU 负载进行自动扩展或者收缩，虚拟机也会自动的在负载均衡器中自动添加或者删除。
- 压力测试用具，可以使用 LoadRunner，Apache AB 等等，在本例中，使用 PHP 产生压力，达到 CPU 阈值要求。

(1) 定义负载均衡器，首先我们需要增加一个负载均衡器的资源，这个资源依赖于公共 IP 地址，即前端 IP 地址。

```
{
  "type": "Microsoft.Network/loadBalancers",
  "name": "[variables('loadBalancerName')]",
  "location": "[resourceGroup().location]",
  "apiVersion": "[variables('networkApiVersion')]",
  "dependsOn": [
    "[concat('Microsoft.Network/publicIPAddresses/',
      variables('publicIPAddressName'))]",
  ],
  "properties": {
    "frontendIPConfigurations": [
      {
        "name": "LoadBalancerFrontEnd",
        "properties": {
```

```
"publicIPAddress": {
  "id": "[variables('publicIPAddressID')]"
}
```

(2) 定义负载均衡规则，前端的请求通过公网 IP 地址或者 DNS 进来，通过默认的地址分发给后端地址池，使用 TCP 协议，前后端均为标准 80 端口号，也可同时设置负载均衡器 `idleTime` 空闲超时时间，最长可设置为 30 分钟；另外针对 HTTP 请求，需要设置一下针对 80 端口的探测，以此判断后端虚拟机是否健康。

```
{
  "loadBalancingRules": [
    {
      "name": "LBRule",
      "properties": {
        "frontendIPConfiguration": {
          "id": "[variables('frontEndIPConfigID')]"
        },
        "backendAddressPool": {
          "id": "[variables('lbPoolID')]"
        },
        "protocol": "tcp",
        "frontendPort": 80,
        "backendPort": 80,
        "enableFloatingIP": false,
        "idleTimeoutInMinutes": 5,
        "probe": {
          "id": "[variables('lbProbeID')]"
        },
        .....
        "probes": [
          {
            "name": "tcpProbe",
            "properties": {
              "protocol": "tcp",
              "port": 80,
              "intervalInSeconds": "5",
              "numberOfProbes": "2"
            },
            .....
          }
        ]
      }
    }
  ]
}
```

(3) Azure 提供了定制化脚本扩展，以方便用户定制化部署，可以让你在虚拟机部署完成后，运行自定义的脚本，安装自己软件和应用，具体的用法如下，你可以将你的应用放在 Azure 存储中，本例中放在了 github 上，然后执行 `bash`，进行安装配置：

```
},
"extensionProfile": {
  "extensions": [
    {
      "name": "lapextension",
      "properties": {
```

```

        "publisher": "Microsoft.OSTCEExtensions",
        "type": "CustomScriptForLinux",
        "typeHandlerVersion": "1.4",
        "autoUpgradeMinorVersion": false,
        "settings": {
            "fileUris": [
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/install_lap.sh",
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/index.php",
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/do_work.php"
            ],
            "commandToExecute": "bash install_lap.sh"
        }
    }
}

```

本次测试中，提供了两个 PHP Web 文件，一个是 index.php，用来显示当前的 Web 应用跑在哪个服务器上，另外一个 do\_work.php 用来给 Web 服务器产生压力，触发自动扩展。

(4) 配置 VMSS 自动扩展规则，例如在什么情况下自动扩展或自动收缩。在本例中，定义 VMSS 中 CPU 负载在过去的 5 分钟内高于 60% 就进行自动扩展，低于 50% 就自动收缩。

```

    "rules": [
        {
            "metricTrigger": {
                "metricName": "\\Processor\\PercentProcessorTime",
                "metricNamespace": "",
                "metricResourceUri": "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/', resourceGroup().name, '/providers/Microsoft.Compute/virtualMachineScaleSets/', variables('namingInfix'))]",
                "timeGrain": "PT1M",
                "statistic": "Average",
                "timeWindow": "PT5M",
                "timeAggregation": "Average",
                "operator": "GreaterThan",
                "threshold": 60.0
            },
            "scaleAction": {
                "direction": "Increase",
                "type": "ChangeCount",
                "value": "1",
                "cooldown": "PT1M"
            }
        },
        {
            "metricTrigger": {
                "metricName": "\\Processor\\PercentProcessorTime",
                "metricNamespace": "",
                "metricResourceUri": "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/', resourceGroup().name, '/providers/Microsoft.Compute/virtualMachineScaleSets/', variables('namingInfix'))]",
                "timeGrain": "PT1M",
                "statistic": "Average",
                "timeWindow": "PT5M",
                "timeAggregation": "Average",
                "operator": "LessThan",
                "threshold": 50.0
            },
            "scaleAction": {
                "direction": "Decrease",
                "type": "ChangeCount",
                "value": "-1",
                "cooldown": "PT1M"
            }
        }
    ]
}

```



```

    },
    "scaleAction": {
      "direction": "Decrease",
      "type": "ChangeCount",
      "value": "1",
      "cooldown": "PT1M"
    }
  }

```

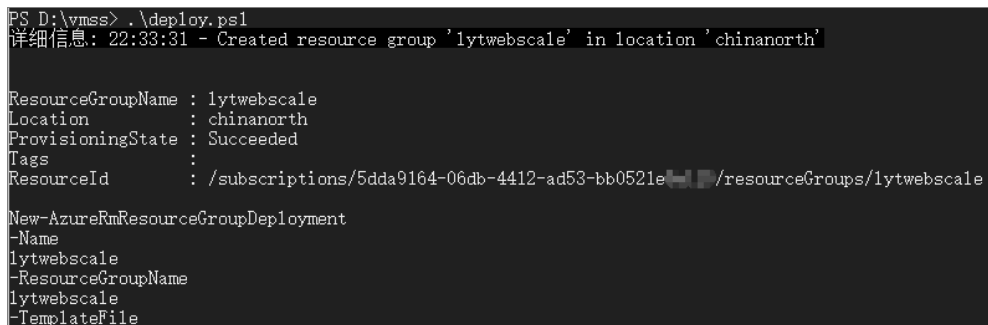
(5) 配置参数文件，定义 VMSS 的名称、初始在 VMSS 中虚拟机数量、用户名和密码。

```

{
  " $schema " : " http://schema.management.azure.com/schemas/2015-01-01/
deploymentParameters.json# ",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmssName": {
      "value": "Webscaleset"
    },
    "instanceCount": {
      "value": 2
    },
    "adminUsername": {
      "value": "XXXXXX"
    },
    "adminPassword": {
      "value": "XXXXXX"
    }
  }
}

```

(6) 最后，我们使用 Powershell 进行部署，如图 9.5-7 所示。



```

PS D:\vmss> .\deploy.ps1
详细信息: 22:33:31 - Created resource group 'lytwebscale' in location 'chinanorth'

ResourceGroupName : lytwebscale
Location           : chinanorth
ProvisioningState  : Succeeded
Tags               :
ResourceId          : /subscriptions/5dda9164-06db-4412-ad53-bb0521e1-1234/resourceGroups/lytwebscale

New-AzureRmResourceGroupDeployment
-Name
lytwebscale
-ResourceGroupName
lytwebscale
-TemplateFile

```

图 9.5-7

(7) 部署完成后，登录 Azure Portal: <https://portal.azure.cn>，可以看到新的 VMSS 集合已经部署成功，包括有一个扩展集，一个负载均衡器，一个公网 IP 地址及多个用于分发 VM 的存储账号，如图 9.5-8 所示。



图 9.5-8

(8) 进入虚拟机扩展集，查看当前实例，可以看到当前有 2 个实例，如图 9.5-9 所示。



图 9.5-9

(9) 打开负载均衡器，获得公网的 IP 地址或者 DNS，在浏览器中打开，可以看到当前连接的是 001 Web 服务器，该页面是一个 demo 页面，用于给虚拟机产生压力；新打开一个浏览器，连接负载均衡器，可以看到请求被分发到了 002 Web 服务器，如图 9.5-10 所示。



图 9.5-10

(10) 在当前的测试页面上，输入 500 秒，作为压力测试时长，单击 "DO work"，那么 PHP 程序就会产生压力，占满 CPU，如图 9.5-11 所示。

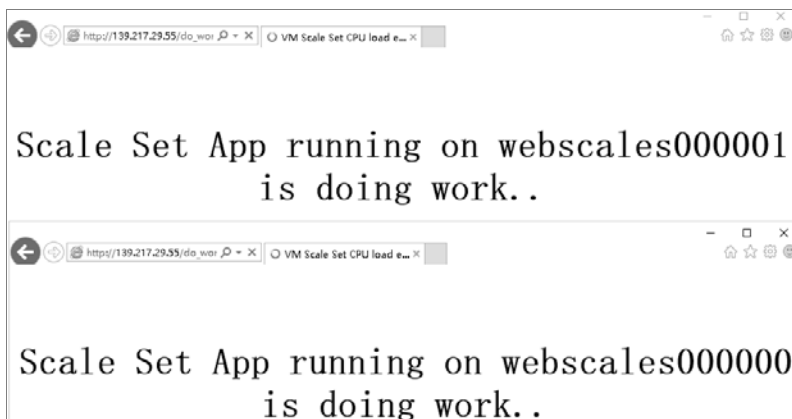


图 9.5-11

(11) CPU 负载连续 5 分钟超过 60% 后，我们打开虚拟机扩展页面的实例项，可以看到，按照之前模板定义的 VMSS 自动扩展规则，虚拟机开始自动增加，如图 9.5-12 所示。

搜索虚拟机实例		
名称	状态	最新模型
<input type="checkbox"/> webscales_0	● 正在运行	是
<input type="checkbox"/> webscales_1	● 正在运行	是
<input type="checkbox"/> webscales_4	● 正在创建 (正在运行)	是
<input type="checkbox"/> webscales_5	● 正在创建 (正在运行)	是

图 9.5-12

(12) 压力测试完成，虚拟机扩展集的压力逐步低于 50%，这个时候，整个虚拟机扩展集会监测最近 5 分钟的负载情况，一旦满足收缩要求，就会执行 cooldown 的过程，逐步移除 Web 服务器，也会从负载均衡器移除，降低成本，如图 9.5-13 所示。

搜索虚拟机实例		
名称	状态	最新模型
<input type="checkbox"/> webscales_0	● 正在运行	是
<input type="checkbox"/> webscales_1	● 正在运行	是
<input type="checkbox"/> webscales_4	● 正在删除	是

图 9.5-13

通过以上实验方案，该企业可以方便的使用 VMSS+ARM 快速的构建自动可扩展的 Web 集群，并且使用定制化脚本部署需要的应用程序。

# 第十章 备 份

除了性能和安全方面的考虑，灾难恢复也是很多用户考虑的一个问题，高可用方案固然可以保证应用的可用性，但是生产环境核心服务的备份和还原也是非常必要的。本章针对 Azure 备份和还原服务的应用场景，备份方法，疑难解答以及排错方法进行了详细说明，通过仔细阅读本章内容，读者可以掌握 Azure 备份还原服务的具体使用方法和排错手段。

## 10.1 Azure 备份功能概述

### 10.1.1 可以备份哪些应用程序和工作负荷

下表提供了可使用 Azure 备份保护的数据和工作负荷的矩阵。Azure 备份解决方案列具有该解决方案部署文档的链接。每个 Azure 备份组件均可在经典（Service Manager 部署）或资源管理器部署模型环境中进行部署。

完成后，使用创建好的系统磁盘创建虚拟机，虚拟机型号可以选择例如 DS，FS 系列的虚拟机。完成创建后，再将创建好的数据磁盘挂载到虚拟机上即可。见表 10.1-1。

表 10.1-1

或工作负荷	源 环 境	Azure 备份解决方案
文件和文件夹	Windows Server	Azure 备份代理、System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）
文件和文件夹	Windows 计算机	Azure 备份代理、System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）
Hyper-V 虚拟机（Windows）	Windows Server	System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）
Hyper-V 虚拟机（Linux）	Windows Server	System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）
Microsoft SQL Server	Windows Server	System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）
Microsoft SharePoint	Windows Server	System Center DPM（带 Azure 备份代理）、Azure 备份服务器（带 Azure 备份代理）

(续表)

或工作负荷	源 环 境	Azure 备份解决方案
Microsoft Exchange	Windows Server	System Center DPM (带 Azure 备份代理)、Azure 备份服务器 (带 Azure 备份代理)
Azure IaaS VM (Windows)	在 Azure 中运行	Azure 备份 (VM 扩展)
Azure IaaS VM (Linux)	在 Azure 中运行	Azure 备份 (VM 扩展)

## 10.1.2 Linux 支持

表 10.1-2 显示了支持 Linux 的 Azure 备份组件，本地 Linux 物理机不支持。

表 10.1-2

组 件	Linux (Azure 认可) 支持
Azure 备份 (MARS) 代理	否 (仅限基于 Windows 的代理)
System Center DPM	在 Hyper-V 和 VMWare 上对 Linux 来宾 VM 进行文件一致性备份 (不适用于 Azure VM) 对 Hyper-V 和 VMWare Linux 来宾 VM 进行 VM 还原
Azure 备份服务器	在 Hyper-V 和 VMWare 上对 Linux 来宾 VM 进行文件一致性备份 (不适用于 Azure VM) 对 Hyper-V 和 VMWare Linux 来宾 VM 进行 VM 还原
Azure IaaS VM 备份	应用程序一致性备份，使用前脚本和后脚本框架还原所有 VM 磁盘 VM

## 10.2 备份 Azure 虚拟机

### 10.2.1 计划之最佳实践

我们建议在为虚拟机配置备份时遵循以下做法：

(1) 请勿计划同时备份同一云服务中的 10 个以上经典 VM。如果要备份同一云服务中的多个 VM，建议将备份开始时间错开一小时。

(2) 请勿计划同时备份 40 个以上 VM。

(3) 将 VM 备份安排在非高峰时间进行。这样备份服务会使用 IOPS 将数据从客户存储账户传输到保管库。

(4) 确保策略在分布于不同存储账户的 VM 上应用。建议不要使用同一备份计划保护单个存储账户中总数超过 20 个的磁盘。如果存储账户中的磁盘超过 20 个，可将这些 VM 分散到多个策略中，以便在备份过程的传输阶段能够获得所需的 IOPS。

(5) 不要将运行在高级存储上的 VM 还原到同一存储账户。如果还原操作过程与备份操作一致，则会减少备份的可用 IOPS。

(6) 建议将每个高级 VM 运行在不同的高级存储账户上，确保优化备份性能。

### 10.2.2 准备之备份和还原 VM 时的限制

- (1) 支持备份拥有 16 个以上数据磁盘的虚拟机。
- (2) 不支持备份使用保留 IP 地址且未定义终结点的虚拟机。
- (3) 备份数据不包括连接到 VM 的网络挂载驱动器。
- (4) 不支持在还原过程中替换现有虚拟机。首先删除现有虚拟机以及任何关联的磁盘，然后从备份还原数据。
- (5) 不支持跨区域备份和恢复。
- (6) Azure 的所有公共区域都支持使用 Azure 备份服务来备份虚拟机（请参阅受支持区域的清单）。在创建保管库期间，如果你要寻找的区域目前不受支持，则不会在下拉列表中显示它。
- (7) 只有特定操作系统版本才支持使用 Azure 备份服务备份虚拟机：
- (8) 仅支持通过 PowerShell 还原属于多 DC 配置的域控制器（DC）VM。阅读有关还原多 DC 域控制器的详细信息。
- (9) 仅支持通过 PowerShell 还原采用以下特殊网络配置的虚拟机。还原操作完成后，在 UI 中使用还原工作流创建的虚拟机将不采用这些网络配置。若要了解详细信息，请参阅还原采用特殊网络配置的 VM。
- (10) 采用负载均衡器配置的虚拟机（内部和外部）。
- (11) 使用多个保留 IP 地址的虚拟机。
- (12) 使用多个网络适配器的虚拟机。

### 10.2.3 管理之审核操作

可以通过 Azure 备份来查看客户触发的备份操作的“操作日志”，因此可以轻松地了解针对备份保管库执行了哪些管理操作。通过操作日志，可以针对备份操作进行很好的事后总结和审核。

操作日志中记录了以下操作：

- (1) 注册
- (2) 注销
- (3) 配置保护
- (4) 备份（二者均可通过 BackupNow 以按需备份的形式进行计划）
- (5) 还原
- (6) 停止保护
- (7) 删除备份数据
- (8) 添加策略
- (9) 删除策略
- (10) 更新策略
- (11) 取消作业

若要查看某个备份保管库的相应操作日志，请执行以下操作：导航到 Azure 门户预览中的“管理服务”，然后单击“操作日志”选项卡，如图 10.2-1 所示。

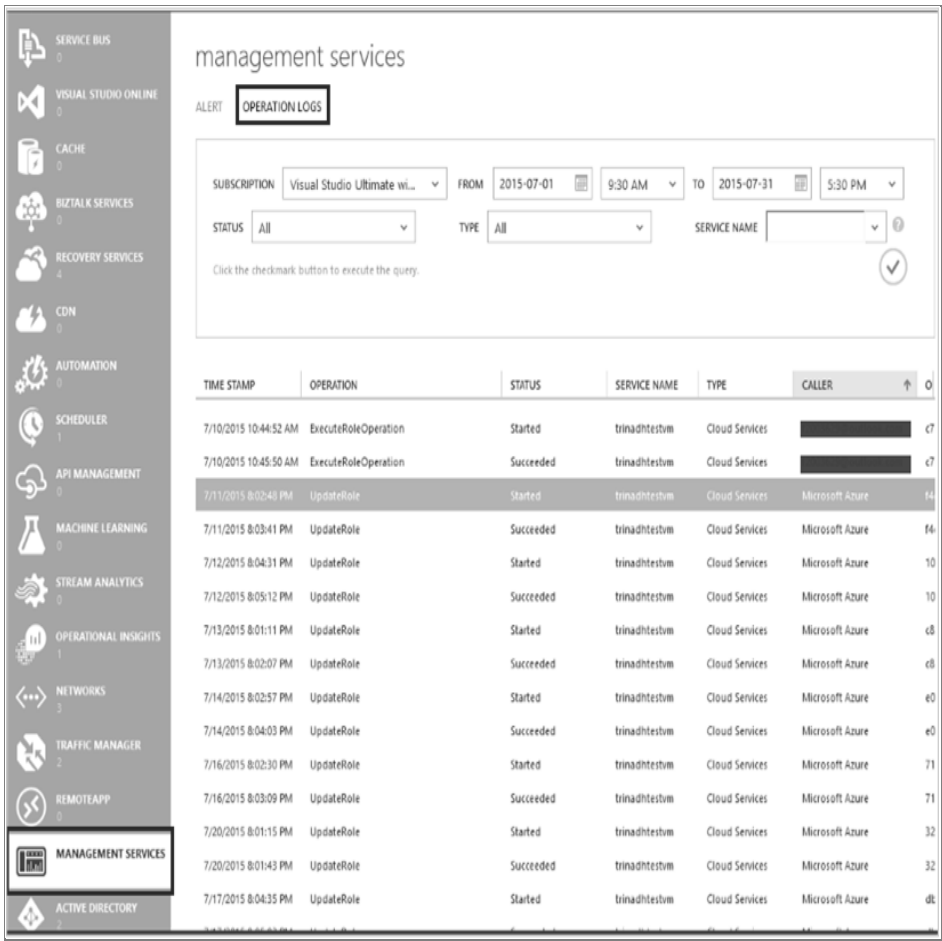


图 10.2-1

在筛选器中选择“备份”作为“类型”，在“服务名称”中指定备份保管库名称，然后单击“提交”，如图 10.2-2 所示。

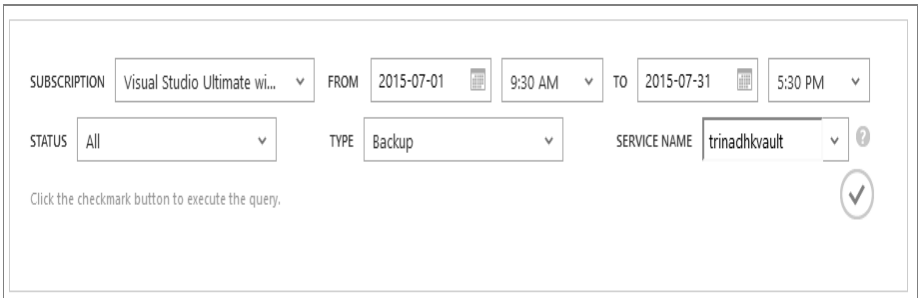


图 10.2-2

在操作日志中，选择任意操作，然后单击“详细信息”查看与操作相对应的详细信息，如图 10.2-3 所示。

ALERT   OPERATION LOGS

SUBSCRIPTIONVisual Studio Ultimate wi...

FROM2015-09-019:30 AM

TO2015-09-095:30 PM

STATUSAll

TYPEBackup

SERVICE NAMEtrinadhivault

Click the checkmark button to execute the query.

TIME STAMP	OPERATION	STATUS	SERVICE NAME	TYPE	CPU	OPERATION ID	
9/4/2015 8:00:47 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		e964152-96d-46d-8...	
9/4/2015 8:22:22 PM	Microsoft.Backup/backup/vault/Backup	Succeeded	trinadhivault	Microsoft.Backup		0e65f696-332d-414d-...	
9/3/2015 8:02:30 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		56896bd-b046-42e4-...	
9/4/2015 1:10:52 AM	Microsoft.Backup/backup/vault/Backup	Succeeded	trinadhivault	Microsoft.Backup		e5721963-9994-4983-a...	
9/1/2015 8:37:30 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		d6d6931-4918-4f54-x...	
9/1/2015 8:40:32 PM	Microsoft.Backup/backup/vault/Backup	Succeeded	trinadhivault	Microsoft.Backup		2a627790-e422-4e5d-b...	
9/5/2015 8:03:57 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		b0fa7000-dec7-42b7-9...	
9/5/2015 8:23:58 PM	Microsoft.Backup/backup/vault/Backup	Succeeded	trinadhivault	Microsoft.Backup		385f523c-2207-4f4d-b...	
9/7/2015 4:31:03 PM	Microsoft.Backup/backup/vault/CreateProte...	Succeeded	trinadhivault	Microsoft.Backup		f8377f5d-644d-4c9d-b...	
9/7/2015 8:04:19 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		5041ac14-41d5-440f-b...	
9/7/2015 8:19:43 PM	Microsoft.Backup/backup/vault/Backup	Succeeded	trinadhivault	Microsoft.Backup		ck1b8f5-fa11-457c-9...	
9/2/2015 5:22:49 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		e45a1b1e-e4d2-4ccb-8...	
9/2/2015 5:22:52 PM	Microsoft.Backup/backup/vault/Backup	Failed	trinadhivault	Microsoft.Backup		dc1d5e96-8f1d-4ac1-8...	
9/2/2015 8:26:35 PM	Microsoft.Backup/backup/vault/Backup	Started	trinadhivault	Microsoft.Backup		b6648f3d-7702-430c-a4...	
9/2/2015 8:26:35 PM	Microsoft.Backup/backup/vault/Backup	Failed	trinadhivault	Microsoft.Backup		10646490-44d3-4885-...	

i

DETAILS

图 10.2-3

“详细信息向导”包含与触发的操作、作业 ID、触发此操作时所在的资源以及操作启动时间相关的信息，如图 10.2-4 所示。

OPERATION DETAILS

Entity Name:  
trinadhtestvm

Job Id:  
53f335e0-3a1b-4f91-84d0-2f3c37e601eb

Microsoft.Resources/EventNameV2:  
Backup

Microsoft.Resources/Operation:  
Microsoft.Backup/backup/vault/Backup

Microsoft.Resources/ResourceUri:  
/subscriptions/73c3de5e-4719-49df-a619-bf779dda2f3b/resourceGroups/RecoveryServices-Q...

Start Time:  
2015-09-01 14:33:28Z

图 10.2-4



### 10.2.4 故障排查之疑难解答

可参考表 10.2-1～表 10.2-3 中所列的信息，排查使用 Azure 备份时遇到的错误。

表 10.2-1 备份

错误详细信息	解决方法
无法执行该操作，因为 VM 已不存在。- 停止保护虚拟机，无需删除备份数据。	<p>当主 VM 已删除，而备份策略仍继续查找用于备份的 VM 时，将会发生这种情况。若要修复此错误，请执行以下操作：</p> <ol style="list-style-type: none"> <li>1. 使用相同的名称和相同的资源组名称[云服务名称] 重新创建虚拟机，（或者）</li> <li>2. 停止保护虚拟机，删除或不删除备份数据。</li> </ol>
无法与 VM 代理通信，因此无法获取快照状态。 - 确保 VM 具有 Internet 访问权限。	<p>如果 VM 代理出现问题，或以某种方式阻止了对 Azure 基础结构的网络访问，则会引发此错误。详细了解如何调试 VM 快照问题。</p> <p>如果 VM 代理未导致任何问题，则重启 VM。有时 VM 状态不正确可能会导致问题，而重新启动 VM 则会重置此“错误状态”</p>
恢复服务扩展操作失败。- 确保虚拟机上有最新的虚拟机代理，并且代理服务正在运行。请重试备份操作，如果失败，请与 Microsoft 支持部门联系。	<p>VM 代理过期会引发此错误。请参阅以下“更新 VM 代理”部分，更新 VM 代理。</p>
虚拟机不存在。- 请确保该虚拟机存在，或选择其他虚拟机。	<p>当主 VM 已删除，而备份策略仍继续查找用于执行备份的 VM 时，会发生这种情况。若要修复此错误，请执行以下操作：</p> <ol style="list-style-type: none"> <li>1. 使用相同的名称和相同的资源组名称 [云服务名称] 重新创建虚拟机，（或者）</li> <li>2. 停止保护虚拟机，无需删除备份数据。</li> </ol>
命令执行失败。- 此项上当前正在进行另一项操作。请等到前一项操作完成，然后重试	<p>VM 的现有备份正在运行，现有作业正在运行时，无法启动新作业。</p>
从备份保管库复制 VHD 超时 - 请在几分钟后重试操作。如果问题持续出现，请联系 Microsoft 支持。	<p>发生这种情况的可能原因包括：存储端出现暂时性错误；或备份服务没有从托管 VM 的存储账户获得足够的 IOPS，无法在超时期限内将数据传输到保管库。设置备份时，请确保遵循最佳做法。尝试将 VM 移到未加载的其他存储账户，然后重试备份。</p>
发生内部错误，备份失败 - 请在几分钟后重试操作。如果问题仍然存在，请联系 Microsoft 支持	<p>导致此错误发生的原因有 2 个：</p> <ol style="list-style-type: none"> <li>1. 访问 VM 存储时发生暂时性问题。请检查 Azure 状态，确定区域中是否存在与计算、存储或网络相关的任何问题。问题解决后，请重试此备份作业。</li> <li>2. 已删除原始 VM，因此无法获取恢复点。若要保留已删除 VM 的备份数据，但要删除备份错误：请取消保护 VM 并选择保留数据选项。此操作会停止计划备份作业和重复错误消息。</li> </ol>
无法在选择的项上安装 Azure 恢复服务扩展 - VM 代理是 Azure 恢复服务扩展的必备组件。 安装 Azure VM 代理并重启注册操作	<ol style="list-style-type: none"> <li>1. 检查是否已正确安装 VM 代理。</li> <li>2. 确保已正确设置 VM 配置中的标志。</li> </ol>
扩展安装失败，出现错误“COM+ 无法与 Microsoft 分布式事务处理协调器通信”。	<p>这通常意味着 COM+ 服务未运行。请与 Microsoft 支持部门联系，以获取解决此问题所需的帮助。</p>
快照操作失败，出现 VSS 操作错误“此驱动器已通过 BitLocker 驱动器加密锁定”。必须通过控制面板解锁此驱动器。	<p>关闭 VM 上所有驱动器的 BitLocker，观察 VSS 问题是否得到解决</p>

(续表)

错误详细信息	解决方法
VM 未处于允许备份的状态。	<ul style="list-style-type: none"> <li>请检查 VM 是否处于“正在运行”和“关闭”之间的暂时性状态中。如果是，请等待 VM 状态变为其中之一，然后再次触发备份。</li> <li>如果 VM 是 Linux VM 并使用[安全性增强的 Linux]内核模块，则需要从安全策略排除 Linux 代理路径(/var/lib/waagent)，确保安装备份扩展。</li> </ul>
找不到 Azure 虚拟机。	<p>如果主 VM 已删除，而备份策略仍继续查找用于执行备份的 VM，则会发生这种情况。若要修复此错误，请执行以下操作：</p> <ol style="list-style-type: none"> <li>使用相同的名称和相同的资源组名称 [云服务名称] 重新创建虚拟机，（或者）</li> <li>禁用对此 VM 的保护，从而不创建备份作业。</li> </ol>
虚拟机上不存在虚拟机代理 - 请安装任何必备组件和 VM 代理，然后重启操作。	请参考如何安装 VM 代理以及如何验证 VM 代理安装。
快照操作失败，因为 VSS 编写器处于错误状态	<p>需重新启动处于错误状态的 VSS（卷影复制服务）编写器。为此，请在提升权限的命令提示符处运行 <code>vssadmin list writers</code>。输出包含所有 VSS 编写器及其状态。对于每个状态不为“[1]稳定”的 VSS 编写器，请在提升权限的命令提示符处运行以下命令，以便重新启动 VSS 编写器</p> <pre>net stop serviceName net start serviceName</pre>
快照操作失败，因为对配置进行分析失败	<p>发生这种情况是因为以下 MachineKeys 目录的权限已更改: %systemdrive%\programdata\microsoft\crypto\rsa\machinekeys</p> <p>请运行以下命令，验证 MachineKeys 目录的权限是否为默认权限：  <code>_icacls %systemdrive%\programdata\microsoft\crypto\rsa\machinekeys_</code></p> <p>默认权限为：  <b>Everyone:(R,W)</b>  <b>BUILTIN\Administrators:(F)</b></p> <p>如果你看到 MachineKeys 目录的权限不同于默认权限，请执行以下步骤来更正权限、删除证书以及触发备份。</p> <ol style="list-style-type: none"> <li>修复 MachineKeys 目录上的权限。        通过目录的“浏览器安全属性”和“高级安全设置”将权限重置回默认值，从目录中删除任何多余的（相对于默认设置）用户对象，确保“Everyone”权限具有下述特殊访问权限：       <ul style="list-style-type: none"> <li>-列出文件夹/读取数据</li> <li>-读取属性</li> <li>-读取扩展的属性</li> <li>-创建文件/写入数据</li> <li>-创建文件夹/追加数据</li> <li>-写入属性</li> <li>-写入扩展的属性</li> <li>-读取权限</li> </ul> </li> <li>删除“颁发对象”字段为“Azure Service Management for Extensions”或“Azure CRP Certificate Generator”的证书。           <ul style="list-style-type: none"> <li>o 打开证书（本地计算机）控制台</li> <li>o 删除“颁发对象”字段为“Azure Service Management for Extensions”或“Azure CRP Certificate Generator”的证书（在“个人”→“证书”下）。</li> </ul> </li> <li>触发 VM 备份。</li> </ol>

(续表)

错误详细信息	解决方法
验证失败, 因为虚拟机仅使用 BEK 进行加密。仅可为同时使用 BEK 和 KEK 进行加密的虚拟机启用备份。	虚拟机应同时使用 BitLocker 加密密钥和密钥加密密钥进行加密。之后, 应启用备份。
虚拟机应同时使用 BitLocker 加密密钥和密钥加密密钥进行加密。之后, 应启用备份。	<p>请尝试启动 Windows 服务“COM + 系统应用程序”(通过权限提升的命令提示符: <code>_net start COMSysApp_</code>)。</p> <p>如果启动失败, 请执行以下步骤:</p> <ol style="list-style-type: none"> <li>1. 验证服务“分布式事务处理协调器”的登录账户是否为“网络服务”。如果不是, 请将其更改为“网络服务”, 重启此服务, 然后尝试启动服务“COM + 系统应用程序”。</li> <li>2. 如果仍然无法启动, 请通过以下步骤卸载/安装服务“分布式事务处理协调器”: <ul style="list-style-type: none"> <li>- 停止 MSDTC 服务</li> <li>- 打开命令提示符 (cmd)</li> <li>- 运行命令“<code>msdtc -uninstall</code>”</li> <li>- 运行命令“<code>msdtc -install</code>”</li> <li>- 启动 MSDTC 服务</li> </ul> </li> <li>3. 启动 Windows 服务“COM + 系统应用程序”, 启动后, 从门户触发备份。</li> </ol>
未能冻结一个或多个 VM 装入点来获取文件系统一致快照	<ol style="list-style-type: none"> <li>1. 使用“<code>tune2fs</code>”_命令检查所有装入设备的文件系统状态。 例如: <code>tune2fs -l /dev/sdb1   grep "Filesystem state"</code></li> <li>2. 使用“<code>umount</code>”_命令卸载文件系统状态不是干净状态的设备</li> <li>3. 使用“<code>fsck</code>”_命令在这些设备上运行 <code>FilesystemConsistency</code> 检查</li> <li>4. 再次装入设备, 并尝试备份。</li> </ol>
快照操作失败, 因为创建安全网络通信通道失败	<ol style="list-style-type: none"> <li>1. 在权限提升模式下运行 <code>regedit.exe</code>, 打开注册表编辑器。</li> <li>2. 标识系统中存在的所有 .NetFramework 版本。它们位于注册表项“<code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft</code>”的层次结构下</li> <li>3. 为注册表项中存在的每个 .NetFramework 添加以下键: " <code>SchUseStrongCrypto</code> " =dword:00000001</li> </ol>
快照操作失败, 因为安装 Visual C++ Redistributable for Visual Studio 2012 失败	<p>导航到 <code>C:\Packages\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot\agentVersion and install vcaredist2012_x64</code>。确保允许安装此服务的注册表项值设置为正确的值, 即注册表项 <code>HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\Msiserver</code> 的值设置为 3, 而不是 4。如果仍然遇到安装问题, 请通过权限提升的命令提示符运行 <code>MSIEXEC /UNREGISTER</code>, 然后运行 <code>MSIEXEC /REGISTER</code>, 重启安装服务。</p>

表 10.2-2 作业

错误详细信息	解决方法
此作业类型不支持取消操作 - 请等待作业完成。	无
作业未处于可取消状态 - 请等待作业完成。 或 所选作业未处于可取消状态 - 请等待作业完成。	作业很可能已接近完成状态。请等待作业完成。
不能取消作业, 因为作业并未处于进行状态 - 只能对正在进行的作业执行取消操作。请尝试取消正在进行的作业。	如果存在临时状态, 则可能会发生这种情况。请稍等片刻, 然后重试取消操作
无法取消作业 - 请等待作业完成。	无

表 10.2-3 还原

错误详细信息	解决方法
发生云内部错误，还原失败	<p>1. 使用 DNS 设置配置了你正在尝试还原的云服务。你可以检查  <code>\$deployment = Get-AzureDeployment -ServiceName "ServiceName" -Slot "Production" Get-AzureDns -DnsSettings \$deployment.DnsSettings</code>          如果配置了地址，则表示配置了 DNS 设置。</p> <p>2. 尝试还原的云服务配置了 ReservedIP，且云服务中现有的 VM 处于停止状态。          可以使用以下 PowerShell cmdlet 检查云服务是否有保留的 IP：  <code>\$deployment = Get-AzureDeployment -ServiceName "servicename" -Slot "Production" \$dep.ReservedIPName</code></p> <p>3. 正在尝试还原同一云服务中具有以下特殊网络配置的虚拟机。</p> <ul style="list-style-type: none"> <li>- 采用负载均衡器配置的虚拟机（内部和外部）</li> <li>- 具有多个保留 IP 的虚拟机</li> <li>- 具有多个 NIC 的虚拟机</li> </ul> <p>请在 UI 中选择新的云服务，或者参阅还原注意事项，了解具有特殊网络配置的 VM</p>
所选 DNS 名称已被使用 - 请指定其他 DNS 名称，然后重试。	<p>此处的 DNS 名称是指云服务名称（通常以 chinacloudapp.cn 结尾）。此名称必须是唯一名称。如果遇到此错误，则需在还原过程中选择其他 VM 名称。</p> <p>此错误仅向 Azure 门户预览用户显示。通过 PowerShell 进行的还原操作会成功，因为它只还原磁盘，不创建 VM。如果在磁盘还原操作之后显式创建 VM，则会遇到该错误。</p>
指定的虚拟网络配置不正确 - 请指定其他虚拟网络配置，然后重试。	无
指定的云服务使用的是保留 IP，这不符合要还原的虚拟机的配置 - 请指定其他不使用保留 IP 的云服务，或者选择其他用于还原的恢复点。	无
云服务已达到输入终结点的数目限制 - 请指定其他云服务或使用现有终结点，重新尝试该操作。	无
备份保管库和目标存储账户位于两个不同的区域 - 请确保还原操作中指定的存储账户与备份保管库位于相同的 Azure 区域。	无
不支持为还原操作指定的存储账户 - 仅支持具有本地冗余或地域冗余复制设置的“基本/标准”存储账户。请选择支持的存储账户	无
针对还原操作指定的存储账户类型不处于在线状态 - 请确保在还原操作中指定的存储账户处于在线状态	Azure 存储中出现暂时性错误或中断时，可能会发生这种情况。请选择另一个存储账户。
已达到资源组配额限制 - 请从 Azure 门户预览中删除某些资源组，或者与 Azure 支持部门联系，请求他们提高限制。	无
所选子网不存在 - 请选择已存在的子网	无
备份服务没有权限访问订阅中的资源。	若要解决这个问题，请先使用选择 VM 还原配置的还原已备份磁盘部分中提到的步骤还原磁盘。之后，使用基于还原的磁盘创建 VM 中提到的 PowerShell 步骤基于还原的磁盘创建完整的 VM。

**Note:** 备份或还原需要一定时间

如果发现备份（超过 12 小时）或还原（超过 6 小时）耗时过长，请确保遵循备份最佳实践。此外，请确保应用程序以最佳方式使用 Azure 存储进行备份。

#### 10.2.4.2 VM 代理

##### 1. 设置 VM 代理

通常，VM 代理已存在于从 Azure 库创建的 VM 中。但是，从本地数据中心迁移的虚拟机上未安装 VM 代理。对于此类 VM，必须显式安装 VM 代理。阅读有关在现有 VM 上安装 VM 代理的详细信息。

对于 Windows VM:

- (1) 下载并安装代理 MSI。需要管理员权限才能完成安装。
- (2) 更新 VM 属性，指明已安装代理。

对于 Linux VM:

- (1) 从 github 安装最新 Linux 代理。
- (2) 更新 VM 属性，指明已安装代理。

##### 2. 更新 VM 代理

- (1) 对于 Windows VM:

更新 VM 代理与重新安装 VM 代理二进制文件一样简单。但是，需要确保在更新 VM 代理时，没有任何正在运行的备份作业。

- (2) 对于 Linux VM:

按照更新 Linux VM 代理上的说明进行操作。我们**强烈建议**只通过分发存储库更新代理。我们不建议直接从 github 下载代理代码并更新。如果最新的代理不可用于用户的分发版，请联系分发版支持人员，获取如何安装最新代理的说明。可在 github 存储库中查找最新 Azure Linux 代理的信息。

#### 10.2.4.3 验证 VM 代理安装

如何检查 Windows VM 上的 VM 代理版本:

- (1) 登录 Azure 虚拟机并导航到 `C:\WindowsAzure\Packages` 文件夹。你应会发现 WaAppAgent.exe 文件已存在。

- (2) 右键单击该文件，转到“属性”，然后选择“详细信息”选项卡。“产品版本”字段应为 2.6.1198.718 或更高。

#### 10.2.4.4 排查 VM 快照问题

VM 备份依赖于向底层存储发出快照命令。如果无法访问存储或者快照任务执行延迟，则可能会导致备份作业失败。以下因素可能会导致快照任务失败。

- (1) 使用 NSG 阻止对存储进行网络访问。

详细了解如何使用 IP 允许列表或通过代理服务器对存储启用网络访问。

- (2) 配置了 Sql Server 备份的 VM 可能会导致快照任务延迟。

默认情况下, VM 备份将在 Windows VM 上发出 VSS 完整备份命令。在运行 SQL Server 且已配置 SQL Server 备份的 VM 上, 这可能会造成快照执行延迟。如果由于快照问题而导致备份失败, 请设置以下注册表项。

```
[HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\BCDRAGENT] "
USEVSSCOPYBACKUP" = " TRUE "
```

(3) 由于在 RDP 中关闭了 VM, VM 状态报告不正确。

如果在 RDP 中关闭了虚拟机, 请返回门户检查是否正确反映了 VM 的状态。如果没有, 请在门户中使用 VM 仪表板上的“关机”选项关闭 VM。

(4) 如果四个以上的 VM 共享相同的云服务, 请配置多个备份策略将备份时间错开, 避免同时启动四个以上的 VM 备份。尝试使策略之间的备份开始时间相差一个小时。

(5) VM 正在以高 CPU/内存使用率运行。

如果虚拟机在运行时的 CPU 或内存使用率很高(超过 90%), 快照任务将排队、延迟并最终超时。在这种情况下, 请尝试进行按需备份。

#### 10.2.4.5 网络排查

与所有扩展一样, 备份扩展也需要访问公共 Internet 才能工作。无法访问公共 Internet 时, 可能会出现以下各种情况:

- (1) 扩展安装可能失败
- (2) 备份操作(如磁盘快照)可能失败
- (3) 显示备份操作状态可能失败

此处说明了在哪些情况下需要解析公共 Internet 地址。需要检查 VNET 的 DNS 配置, 并确保可以解析 Azure URI。

正确完成名称解析后, 还需要提供对 Azure IP 的访问权限。若要取消阻止对 Azure 基础结构的访问, 请执行以下步骤之一:

##### 1. 将 Azure 数据中心 IP 范围加入允许列表

- (1) 获取要列入允许列表的 Azure 数据中心 IP 列表。
- (2) 使用 New-NetRoute cmdlet 取消阻止 IP。在 Azure VM 上提升权限的 PowerShell 窗口中运行此 cmdlet (以管理员身份运行)。
- (3) 向 NSG 添加规则(如果已创建规则), 以允许访问这些 IP。

##### 2. 为 HTTP 流量创建路径

- (1) 如果你指定了某种网络限制(例如网络安全组), 请部署 HTTP 代理服务器来路由流量。可在此处找到部署 HTTP 代理服务器的步骤。
- (2) 向 NSG 添加规则(如果已创建规则), 以允许从 HTTP 代理访问 INTERNET。

##### 3. 用户必须启用 DHCP 才能正常进行 IaaS VM 备份

如果需要静态专用 IP 地址, 你应该通过平台配置该 IP。VM 内的 DHCP 选项应保持启用。查看有关设置静态内部专用 IP 的详细信息。

# 第十一章 自动化运维

随着用户生产系统功能逐渐丰富，在 Azure 环境中部署的资源也越来越多，规模越来越庞大。传统的通过人工部署和管理的方法对于规模较小的生产环境还勉强可以接受，但是对于大规模环境，人工方法一方面耗时耗力，另一方面人工操作的准确度和熟练度难以保证。因此，针对此类需求，本章详细介绍了通过 Azure Powershell，Azure CLI（跨平台命令行），Azure Automation 服务以及 Azure 资源管理器模板对 Azure 资源进行部署，管理的方法，利用这些工具，用户可以实现非常复杂的脚本和模板，一方面加速了部署速度，降低了人力操作的时间成本，另一方面，通过对脚本和模板的复用也可以很轻松地在不同项目中配置类似环境。

## 11.1 Azure Powershell 的安装与使用

### 11.1.1 什么是 Azure PowerShell

Azure PowerShell 是一组模块，提供用于通过 Windows PowerShell 管理 Azure 的 cmdlet。你可以使用 cmdlet 来创建、测试、部署和管理通过 Azure 平台传送的解决方案和服务。在大多数情况下，这些 cmdlet 可用于执行在 Azure 经典管理门户和门户预览中可以执行的任务，例如，创建和配置云服务、虚拟机、虚拟网络和 Web 应用。

### 11.1.2 Azure Powershell 的安装

首先通过下载链接下载最新版本的 Microsoft Azure Powershell 安装工具：

<https://azure.microsoft.com/en-us/documentation/articles/powershell-install-configure/>

如图 11.1-1 所示。

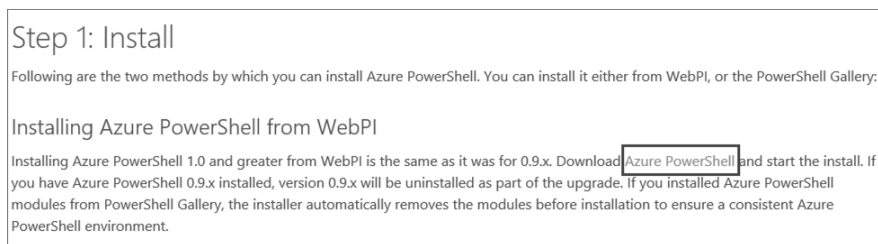


图 11.1-1

最新的 Microsoft Azure Powershell 在安装完成后，不会在系统中生成一个单独的快捷方式 Microsoft Azure Powershell，而仅仅是在系统中安装了 Azure Powershell 的模块和组件。

所以需要打开 Windows Powershell，并执行以下命令：

```
PS C:\> Import-Module -Name Azure
```

使用 Get-Module 命令确认 Azure Powershell 模块已经导入：

```
PS C:\> Get-Module
```

ModuleType	Version	Name	ExportedCommands
Manifest	3.0.0	Azure	{Add-AzureAccount
Manifest	2.2.0	Azure.Storage	{Get-AzureStorage
Manifest	2.2.0	AzureRM.Compute	{Add-AzureRmConta
Manifest	2.2.0	AzureRM.Profile	{Add-AzureRmAccou
Manifest	3.1.0.0	Microsoft.PowerShell.Management	{Add-Computer, Ad
Manifest	3.1.0.0	Microsoft.PowerShell.Utility	{Add-Member, Add-

### 11.1.3 Azure Powershell 的使用

#### 11.1.3.1 登录经典模式（ASM）的方法

使用 Add-AzureAccount -Environment AzureChinaCloud 命令登录 ASM：

```
PS C:\> Add-AzureAccount -Environment AzureChinaCloud
```

回车后，会弹出认证界面，填写您的登录账号，如图 11.1-2 所示。



图 11.1-2

登录成功后，使用 Select-AzureSubscription -SubscriptionName XXXXXX -Default 选择默认订阅：

使用 Get-AzureSubscription -Default 查看默认订阅是否设置成功。



```
PS C:\> Get-AzureSubscription -Default

SubscriptionId      : XXXXXXX
SubscriptionName    : XXXXXXX
Environment         : AzureChinaCloud
DefaultAccount      : XXXXXXX
IsDefault           : True
IsCurrent           : True
TenantId           : XXXXXXX
CurrentStorageAccountName : jonorthstore1
```

选择完默认订阅后，就可以进行各种配置了，比如使用 `Get-AzureVM` 查看当前订阅下虚拟机列表：

```
PS C:\> Get-AzureVM

ServiceName  Name          Status
-----
DanEastCS    CentOS68      StoppedDeallocated
DanEastCS    Dan68Test     StoppedDeallocated
DanEastCS    DanCentOS68   StoppedDeallocated
DanEastCS    DanCentOS73   StoppedDeallocated
DanEastCS    DanSSHDemo    StoppedDeallocated
DanNorthCS   DanServer08N  StoppedDeallocated
.....
```

### 11.1.3.2 登录门户预览模式（ARM）的方法

使用下面的命令登录 Azure 账号：

```
PS C:\> Login-AzureRmAccount -EnvironmentName AzureChinaCloud
```

回车后，会弹出认证界面，填写您的登录账号，如图 11.1-3 所示。



图 11.1-3

登录成功后，使用 `Select-AzureRmSubscription -SubscriptionName XXXXXX` 选择默认订阅：

```
PS C:\> Select-AzureRmSubscription -SubscriptionName Internal-005

Environment      : AzureChinaCloud
Account          : XXXXXX
TenantId         : XXXXXX
SubscriptionId    : XXXXXX
SubscriptionName  : XXXXXX
CurrentStorageAccount :
```

选择完成后，使用 `Get-AzureRmVM` 查看当前订阅下 ARM 虚拟机的列表：

```
PS C:\> Get-AzureRmVM
.....
viders/Microsoft.Compute/virtualMachines/appgateway2vm1
Name                        : appgateway2vm1
Location                   : chinanorth
Tags                       : {}
viders/Microsoft.Compute/availabilitySets/APPGATEWAY2HA
DiagnosticsProfile         :
  BootDiagnostics           :
    Enabled                 : True
    StorageUri              :
https://paularmnorthdiag.blob.core.chinacloudapi.cn/
HardwareProfile            :
  VmSize                   : Basic_A1
NetworkProfile             :
  NetworkInterfaces[0]     :
providers/Microsoft.Network/networkInterfaces/appgateway2vm1326
OSProfile                  :
  ComputerName             : appgateway2vm1
  AdminUsername            : mkbian
  LinuxConfiguration       :
    DisablePasswordAuthentication : False
ProvisioningState          : Succeeded
StorageProfile             :
  ImageReference           :
    Publisher              : OpenLogic
    Offer                  : CentOS
    Sku                    : 6.7
    Version                : latest
OsDisk                    :
  OsType                   : Linux
  Name                     : appgateway2vm1
  Vhd                     :
.....
```

## 11.2 跨平台命令行的安装和使用

### 11.2.1 跨平台命令行 cli 的安装和使用

Azure CLI 是一组开源且跨平台的命令，可以用于管理 Azure 资源。Azure CLI 可以在 linux、mac 和 windows 系统上使用。

本文使用 Centos7.2 为例：

yum 安装 npm,node.js,然后使用安装 azure-cli(注意需要 root 权限或者使用 sudo 命令)：

```
yum update
yum upgrade -y
yum install epel-release
yum install nodejs
yum install npm
npm install -g azure-cli
```

安装完成后，使用命令 azure 查看是否成功：

```
[root@jorgcentos721 ~]# azure
info:
info:      _ _ _ _ _
info:    /_ \  |_ /  | | | _ \ _ |
info:   _ _ / _ \_ / / | | | / _ | _ _
info:  (___ /_ / \_ \_ / \_ / | | \_ | ___)
info:   (____ _ _ )   _ _ _ _ ) _ _
info:      (____ _ _ ) (____ _ _ )
info:
info:  Microsoft Azure: Microsoft's Cloud Platform
info:
info:  Tool version 0.10.8
help:
help:  Display help for a given command
help:    help [options] [command]
help:
help:  Log in to an Azure subscription using Active Directory or a Microsoft
account identity.
help:    login [options]
help:
help:  Log out from Azure subscription using Active Directory. Currently,
the user can log out only via Microsoft organizational account
help:    logout [options] [username]
help:
help:  Open the portal in a browser
help:    portal [options]
help:
help:  Manages the data collection preference.
help:    telemetry [options]
help:
```

```

help:    Commands:
help:    account      Commands to manage your account information and
publish settings
help:    acs          Commands to manage your container service.
help:    ad           Commands to display Active Directory objects
help:    appserviceplan Commands to manage your Azure appserviceplans
help:    availset      Commands to manage your availability sets.
help:    batch         Commands to manage your Batch objects
help:    cdn           Commands to manage Azure Content Delivery Network (CDN)
help:    config        Commands to manage your local settings
help:    datalake       Commands to manage your Data Lake objects
help:    feature        Commands to manage your features
help:    group          Commands to manage your resource groups
help:    hdinsight      Commands to manage HDInsight clusters and jobs
help:    insights       Commands related to monitoring Insights (events,
alert rules, autoscale settings, metrics)
help:    iotHub         Commands to manage your Azure IoT hubs
help:    keyvault       Commands to manage key vault instances in the Azure
Key Vault service
help:    lab           Commands to manage your DevTest Labs
help:    location       Commands to get the available locations
help:    network        Commands to manage network resources
help:    policy          Commands to manage your policies on ARM Resources.
help:    powerbi        Commands to manage your Azure Power BI Embedded
Workspace Collections
help:    provider       Commands to manage resource provider registrations
help:    quotas         Command to view your aggregated Azure quotas
help:    redisCache     Commands to manage your Azure Redis Cache(s)
help:    resource       Commands to manage your resources
help:    role            Commands to manage role definitions
help:    servermanagement Commands to manage Azure Server Managment
resources
help:    servicefabric  Commands to manage your Azure Service Fabric
help:    storage         Commands to manage your Storage objects
help:    tag             Commands to manage your resource manager tags
help:    usage           Command to view your aggregated Azure usage data
help:    vm             Commands to manage your virtual machines
help:    vmss            Commands to manage your virtual machine scale sets.
help:    vmssvm         Commands to manage your virtual machine scale set vm.
help:    Webapp         Commands to manage your Azure Webapps
help:
help:    Options:
help:    -h, --help      output usage information
help:    -v, --version    output the application version
help:
help:    Current Mode: arm (Azure Resource Management)

```

安装成功后，使用 `azure login` 命令来登录您的账号：

命令格式：`azure login -u <azure account name> -p <password> -e AzureChinaCloud`

运行截图：

```
[root@jorgcentos721 ~]#
[root@jorgcentos721 ~]# azure login -u XXXXXX -e AzureChinaCloud
info: Executing command login
Password: *****
/info: Added subscription XXXXXX
info: Added subscription XXXXXX
info: Added subscription XXXXXX
info: Added subscription XXXXXX
info: Added subscription XXXXXX
info: Added subscription XXXXXX
info: Added subscription Internal Consumption
+
info: login command OK
```

使用 `azure config mode <asm/arm>` 来切换您当前使用的模式是经典模式还是门户预览的模式：

```
[root@jorgcentos721 ~]# azure config mode asm
info: Executing command config mode
info: New mode is asm
info: config mode command OK
[root@jorgcentos721 ~]# azure config mode arm
info: Executing command config mode
info: New mode is arm
info: config mode command OK
```

如果您有多个订阅，可以使用 `azure account set <订阅 ID>` 来设置您要使用的订阅（即虚拟机所在的订阅）：

```
[root@jorgcentos721 ~]# azure account set XXXXXX
info: Executing command account set
info: Setting subscription to " XXXXXX " with id " XXXXXX ".
info: Changes saved
info: account set command OK
```

使用 `azure account show` 查看您当前使用的订阅：

```
[root@jorgcentos721 ~]# azure account show
info: Executing command account show
data: Name :
data: ID : XXXXXX
data: State : Enabled
data: Tenant ID : XXXXXX
data: Is Default : true
data: Environment : AzureChinaCloud
data: Has Certificate : No
data: Has Access Token : Yes
data: User name : XXXXXX
info: account show command OK
```

使用 `azure vm list` 查看您的虚拟机列表：

```
[root@jorgcentos721 ~]# azure vm list
info:    Executing command vm list
+ Getting virtual machines
data:    ResourceGro Name          Provisioni PowerState      .....
data:    -----
data:    AATEST      MV1          Succeeded  VM deallocated
data:    CENTOSBIAN  CentOSBian  Succeeded  VM running
data:    CENTOSBIAN  vmformkbian Succeeded  VM deallocated
data:    CENTOSBIAN  win2012forchen Succeeded  VM deallocated
```

使用 `azure vm show --resource-group <资源组名称> --name <虚拟机名称>` 查看具体虚拟机的详细信息：

```
[root@jorgcentos721 ~]# azure vm show --resource-group jorg --name jo-deb8
info:    Executing command vm show
+ Looking up the VM "jo-deb8"
+ Looking up the NIC "jo-deb8879"
+ Looking up the public ip "jo-deb8-ip"
data:    Id                      : XXXXXX
data:    ProvisioningState          : Succeeded
data:    Name                       : jo-deb8
data:    Location                   : chinanorth
data:    Type                       : Microsoft.Compute/virtualMachines
data:
data:    Hardware Profile:
data:      Size                     : Standard_D1
data:
data:    Storage Profile:
data:      Image reference:
data:        Publisher              : credativ
data:        Offer                  : Debian
data:        Sku                    : 8
data:        Version                : latest
data:
data:    OS Disk:
data:      OSType                   : Linux
data:      Name                     : jo-deb8
data:      Caching                  : ReadWrite
.....
```

## 11.2.2 跨平台命令行 cli2.0 preview 的安装和使用

Azure CLI 2.0 Azure 新开放的一组开源且跨平台的命令，用于管理 Azure 资源。

Azure CLI 2.0 可以在 Linux、Mac 和 windows 系统上使用。

测试使用的是 centos7.2 版本的虚拟机。

请注意 Python 2.7.5 要事先安装好，具体安装命令如下。

```
sudo yum check-update;
sudo yum install -y gcc libffi-devel python-devel openssl-devel
curl -L https://aka.ms/InstallAzureCli | bash
```

安装时间大概是 10 分钟左右, Cli 2.0 使用是 az 命令, 和之前的 Cli 使用 azure 命令有一些区别。可以输入 az 查看安装是否成功:

```
[root@jorgcentos721 ~]# az

  /\
 /  \    _____
/    \  | | | | | | | \_/_ \
/      \ | | | | | | |  _/
/_/      \ \ \ \ \ \ \ \ \ \

Welcome to the cool new Azure CLI!

Here are the base commands:

    account    : Manage subscriptions.
    acs        : Manage Azure Container Services.
    ad         : Synchronize on-premises directories and manage Azure Active
Directory resources.
    appservice: Manage your Azure Web apps and App Service plans.
    batch      : Manage Azure Batch.
    cloud      : Manage the registered Azure clouds.
    component  : Manage and update Azure CLI 2.0 (Preview) components.
    configure  : Configure Azure CLI 2.0 Preview or view your configuration.
The command is
                interactive, so just type `az configure` and respond to the
prompts.
    container  : Set up automated builds and deployments for multi-container
Docker applications.
    disk       : Manage Azure Managed Disks.
    documentdb: Manage your Azure DocumentDB (NoSQL) database accounts.
    feature    : Manage resource provider features, such as previews.
    feedback   : Loving or hating the CLI? Let us know!
    group      : Manage resource groups.
    image      : Manage custom Virtual Machine Images.
    iot        : Connect, monitor, and control millions of IoT assets.
    keyvault   : Safeguard and maintain control of keys, secrets, and
certificates.
    lock       :
    login      : Log in to access Azure subscriptions.
    logout     : Log out to remove access to Azure subscriptions.
    network    : Manages Azure Network resources.
    policy     : Manage resource policies.
    provider   : Manage resource providers.
    redis      : Access to a secure, dedicated cache for your Azure
```

```

applications.
    resource : Manage Azure resources.
    role      : Use role assignments to manage access to your Azure resources.
    snapshot  : Manage Azure Snapshots.
    sql       : Manage Azure SQL databases.
    storage   : Durable, highly available, and massively scalable cloud
storage.
    tag       : Manage resource tags.
    vm        : Provision Linux or Windows virtual machines in seconds.
    vmss      : Create highly available, auto-scalable Linux or Windows
virtual machines.

```

首先，登录中国区 azure 之前，需要使用下面的命令进行登录区域的切换：

az cloud set --name “AzureChinaCloud”

```
[root@jorgcentos721 ~]# az cloud set --name AzureChinaCloud
```

登录命令：

az login -u “账号名”

```

[root@jorgcentos721 ~]# az login -u XXXXXX
Password:
[
  {
    "cloudName": "AzureChinaCloud",
    "id": " XXXXXX ",
    "isDefault": false,
    "name": " XXXXXX ",
    "state": "Enabled",
    "tenantId": " XXXXXX ",
    "user": {
      "name": " XXXXXX ",
      "type": "user"
    }
  },

```

改变当前订阅：

az account set --subscription “订阅名称”。

```

[root@jorgcentos721 ~]# az account set --subscription XXXXXX
[root@jorgcentos721 ~]# az account show
{
  "environmentName": "AzureChinaCloud",
  "id": " XXXXXX ",
  "isDefault": false,
  "name": " XXXXXX ",
  "state": "Enabled",
  "tenantId": " XXXXXX ",
  "user": {
    "name": " XXXXXX ",

```



```
" type " : " user "
}
},
```

至此就可以使用 azure cli 2.0 进行具体操作了,但是目前 cli 2.0 还处于 preview 的状态,有很多功能在 mooncake azure 还不支持。

如果其他 linux 发行版上的安装方法,可以参考如下链接:

<https://docs.microsoft.com/en-us/cli/azure/install-az-cli2>

### 11.3 Azure Automation 的配置和使用

借助 Azure 自动化,用户可以自动完成通常要在云环境和企业环境中执行的手动、长时间进行、易出错且重复性高的任务。它可以节省时间,可以提高常规管理任务的可靠性,甚至可以将这些任务安排成按特定的时间间隔自动执行。你可以使用 Runbook 实现这些过程的自动化。本文介绍如何实现自动开关机。

首先,我们需要创建一个自动化账号(Automation Account),如图 11.3-1 所示。

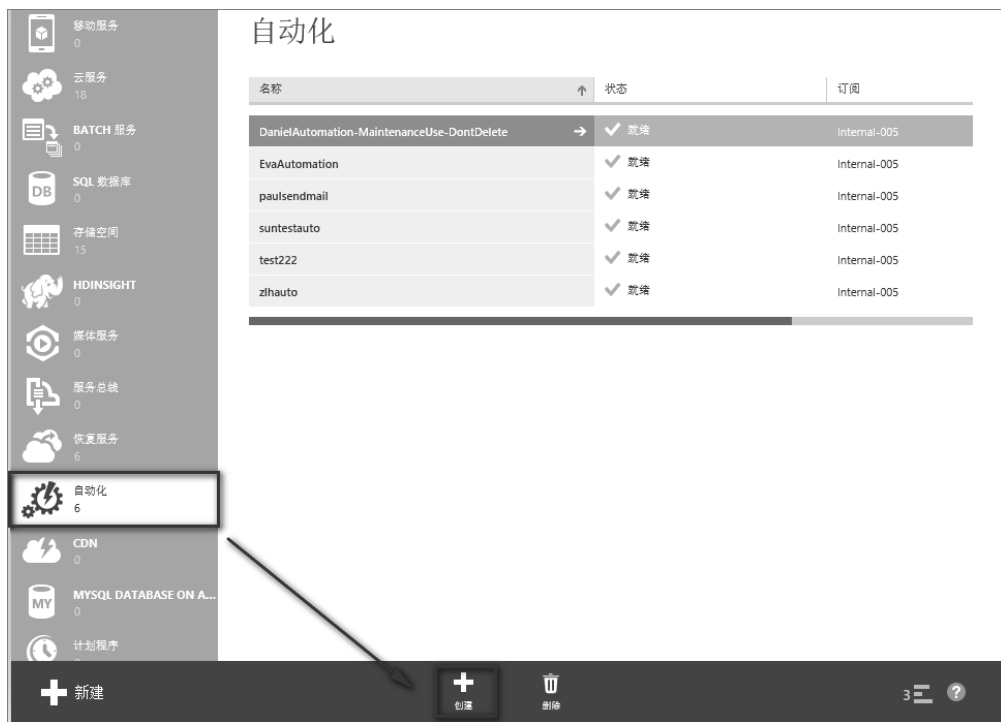


图 11.3-1

在弹出的界面中填写自动化账号名称(用户随便定义一个即可),选择区域以及订阅,如图 11.3-2 所示。



图 11.3-2

完成后就能够在列表中看到这个自动化账号，接着单击左下角“创建”按钮创建一个 Runbook，如图 11.3-3 所示。

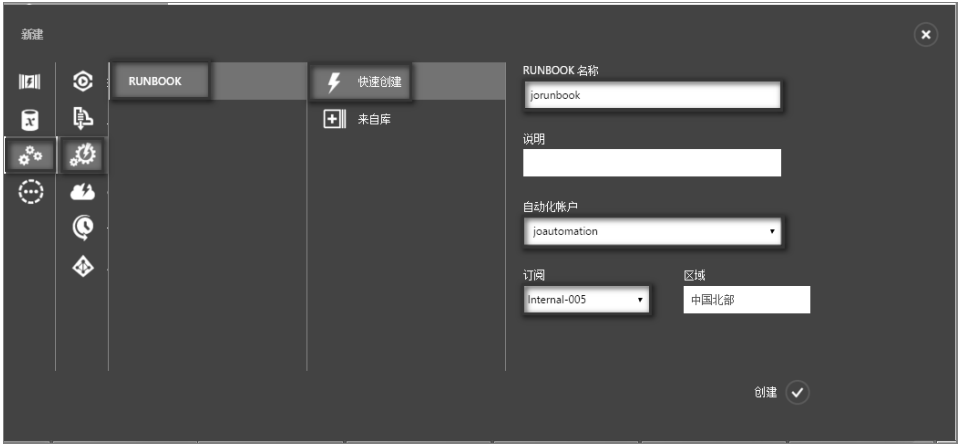


图 11.3-3

创建完成后单击进入这个账号，如图 11.3-4 所示。

自动化				
名称	状态	订阅	位置	
DanielAutomation-MaintenanceUse-DontDelete	✓就绪	Internal-005	中国北部	
EvaAutomation	✓就绪	Internal-005	中国北部	
joautomation	✓就绪	Internal-005	中国北部	→
paulsendmail	✓就绪	Internal-005	中国北部	
suntestauto	✓就绪	Internal-005	中国北部	
test222	✓就绪	Internal-005	中国北部	→
zlhauto	✓就绪	Internal-005	中国北部	

图 11.3-4

可以看到刚刚创建的 Runbook，如图 11.3-5 所示。



图 11.3-5

接着我们在资产选项中添加一个 Credential，如图 11.3-6 所示。

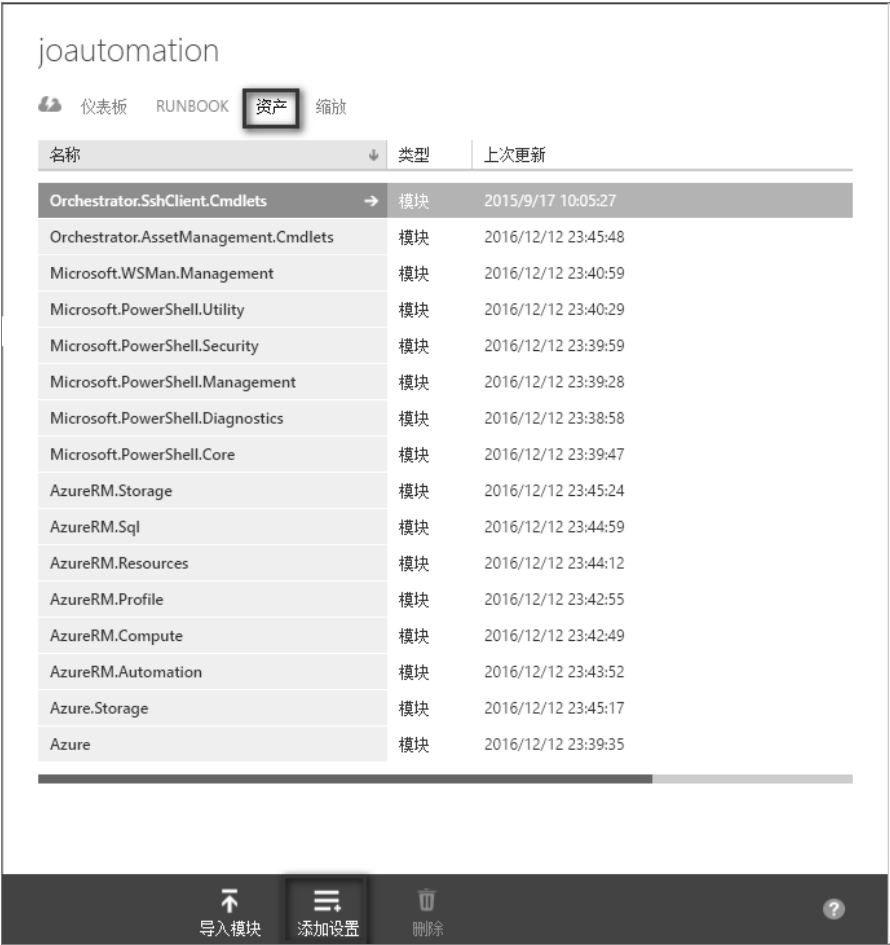


图 11.3-6

在弹出的界面中选择添加凭据，如图 11.3-7 所示。



图 11.3-7

填写凭据类型以及名称，如图 11.3-8 所示。



图 11.3-8

填写用到的用户名和密码（这里是管理 Azure 的用户名和密码），如图 11.3-9 所示。

图 11.3-9

完成后保存。

接着进入之前创建的 Runbook 中，切换到创作选项卡，如图 11.3-10 所示。

图 11.3-10

在草稿编辑界面中输入下面的代码：

```
workflow 您创建的 runbook 名
{
    $Cred = Get-AutomationPSCredential -Name " 您创建的凭据 " ;
    Add-AzureAccount -Credential $Cred -Environment AzureChinaCloud;
    Select-AzureSubscription -SubscriptionName " 您的订阅号 " ;

    Start-AzureVM -ServiceName " 虚拟机云服务名 " -Name " 虚拟机名 " ;
}
```

创建完成后我们单击 PUBLISH，将这段脚本发布为正式版本，如图 11.3-11 所示。



图 11.3-11

注：这里也可以单击 **TEST**（测试）先测试一下脚本的执行情况，确认无误后再单击 **PUBLISH**。

发布完成后，我们在 **PUBLISHED** 选项卡中看到发布的正式脚本，可以单击 **START**（启动）执行这个脚本，如图 11.3-12 所示。

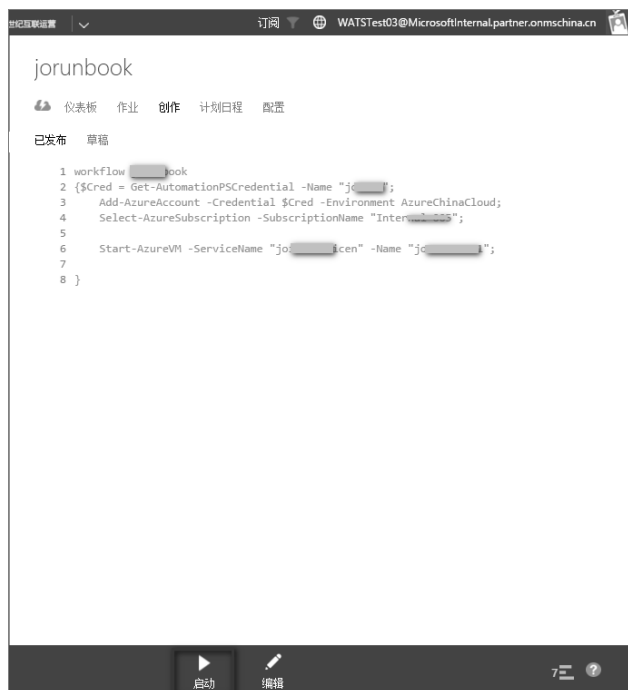


图 11.3-12

执行完成后，会对应的生成一个作业，这里我们脚本测试了一次，所以有一条作业记录，如图 11.3-13 所示。



图 11.3-13

单击作业右侧的白色箭头可以进入到作业中查看具体的执行情况和输出结果。作业执行结束后，可以看到我们的虚拟机已经成功启动了。接着我们为 RUNBOOK 添加一个计划日程，如图 11.3-14 所示。



图 11.3-14

选择配置计划日程，填写一个计划名称，如图 11.3-15 所示。

添加计划日程

### 配置计划日程

名称

josche

说明

→ 2

图 11.3-15

设置一个执行周期，如图 11.3-16 所示。

添加计划日程

### 配置计划日程

类型

一次 每小时 每天

开始时间

2017-03-25 8:50

☒ 计划日程过期时间

2017-04-01 8:50

重复间隔(天数)

1

1

← ✓

图 11.3-16



例如这里根据上面截图中的设置, 这个 RUNBOOK 会在 2017 年 3 月 25 日到 2015 年 4 月 1 日每天的 08:50 执行。如果不希望设置过期时间, 可以取消 SCHEDULE EXPIRES ON 的勾选。

这样我们的开机脚本就设置完成了, 同样的原理, 可以使用下面的脚本配置一个关机脚本:

```
workflow Runbook 名称
{
    $Cred = Get-AutomationPSCredential -Name "凭据名";
    Add-AzureAccount -Credential $Cred -Environment AzureChinaCloud;
    Select-AzureSubscription -SubscriptionName "订阅名";

    Stop-AzureVM -ServiceName "云服务名" -Name "虚拟机名" -Force;
}
```

需要注意的是, 这里 Stop-AzureVM 这个命令一定要添加-Force 参数, 不然在命令执行的时候会停在确认是否要关闭虚拟机的界面而无法完成关闭操作。

后面的配置操作与前面 Start 的类似, 这里就不赘述了。

## 11.4 Azure 资源管理器模板的使用

应用程序的基础结构通常由许多组件构成: 可能有虚拟机、存储账户和虚拟网络, 或 Web 应用、数据库、数据库服务器和第三方服务。这些组件不会以独立的实体出现, 而是以单个实体的相关部件和依赖部件出现。如果你希望以组的方式部署、管理和监视这些组件, 那么, 你可以使用 Azure 资源管理器以组的方式处理解决方案中的资源。可以通过一个协调的操作为解决方案部署、更新或删除所有资源。可以使用一个模板来完成部署, 该模板适用于不同的环境, 例如测试、过渡和生产。资源管理器提供安全、审核和标记功能, 以帮助你在部署后管理资源。

### 11.4.1 资源管理器的基本说明

资源管理器的相关术语

(1) 资源, 可通过 Azure 获取的可管理项。部分常见资源包括虚拟机、存储账户、Web 应用、数据库和虚拟网络, 但这只是其中一小部分。

(2) 资源组, 一个容器, 用于保存 Azure 解决方案的相关资源。资源组可以包含解决方案的所有资源, 也可以只包含想要作为组来管理的资源。根据对组织有利的原则, 决定如何将资源分配到资源组。请参阅资源组。

(3) 资源提供程序, 一种服务, 提供可以通过 Resource Manager 进行部署和管理的资源。每个资源提供程序提供用于处理所部署资源的操作。部分常见资源提供程序包括 Microsoft.Compute (提供虚拟机资源)、Microsoft.Storage (提供存储账户资源) 和 Microsoft.Web (提供与 Web 应用相关的资源)。请参阅资源提供程序。

(4) Resource Manager 模板, 一个 JavaScript 对象表示法 (JSON) 文件, 用于定义一

个或多个要部署到资源组的资源。它也会定义所部署资源之间的依赖关系。使用模板能够以一致方式反复部署资源。请参阅模板部署。

(5) 声明性语法，一种语法，允许你声明“以下是我想要创建的项目”，而不需要编写一系列编程命令来进行创建。**Resource Manager** 模板便是声明性语法的其中一个示例。在该文件中，你可以定义要部署到 **Azure** 的基础结构的属性。

使用资源管理器的优势

- (1) 可以以组的形式部署、管理和监视解决方案的所有资源，而不是单独处理这些资源。
- (2) 可以在整个开发生命周期内重复部署解决方案，并确保以一致的状态部署资源。
- (3) 可以通过声明性模板而非脚本来管理基础结构。
- (4) 可以定义各资源之间的依赖关系，以便按正确的顺序进行部署。
- (5) 可以将访问控制应用到资源组中的所有服务，因为基于角色的访问控制 (RBAC) 已在本机集成到管理平台。
- (6) 可以将标记应用到资源，以逻辑方式组织订阅中的所有资源。
- (7) 可以通过查看一组共享相同标记的资源的成本来明确组织的帐单。

## 11.4.2 模板部署虚拟机

**Azure** 资源管理器模板可让你通过定义资源之间的依赖关系，使用 **JSON** 语言以声明方式指定 **Azure IaaS** 基础结构。使用 **Resource Manager** 可以创建一个模板，用于定义 **Azure** 解决方案的基础结构和配置。使用模板可以在解决方案的整个生命周期内重复部署该解决方案，确保以一致的状态部署资源。从门户创建解决方案时，该解决方案将自动包含部署模板。你无需从头开始创建模板，因为你可以从解决方案的模板着手，并根据你的特定需求自定义该模板。可以通过导出资源组的当前状态或查看特定部署所用的模板，来检索现有资源组的模板。查看导出的模板是了解模板语法的有效方法。

**Resource Manager** 处理模板的方式与处理其他任何请求一样（请参阅一致的管理层中的图像）。它会分析模板，并将其语法转换为相应资源提供程序所需的 **REST API** 操作。例如，当 **Resource Manager** 收到具有以下资源定义的模板时：

```
"resources": [
  {
    "apiVersion": "2016-01-01",
    "type": "Microsoft.Storage/storageAccounts",
    "name": "mystorageaccount",
    "location": "chinanorth",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {
    }
  }
]
```

它会将该定义转换为以下 REST API 操作，然后，该操作将发送到 Microsoft.Storage 资源提供程序：

```
PUT
https://management.chinacloudapi.cn/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Storage/storageAccounts/mystorageaccount?api-version=2016-01-01
REQUEST BODY
{
  "location": "chinanorth",
  "properties": {
  },
  "sku": {
    "name": "Standard_LRS"
  },
  "kind": "Storage"
}
```

模板和资源组的定义方式全由用户决定，解决方案的管理方式也是如此。例如，可以通过单个模板在单个资源组中部署三层式应用程序，如图 11.4-1 所示。

但是，无需在单个模板中定义整个基础结构。通常，合理的做法是将部署要求划分成一组有针对性的模板。可以轻松地将这些模板重复用于不同的解决方案。若要部署特定的解决方案，请创建链接所有所需模板的主模板。图 11.4-1 显示如何通过包含三个嵌套模板的父模板部署三层式解决方案。

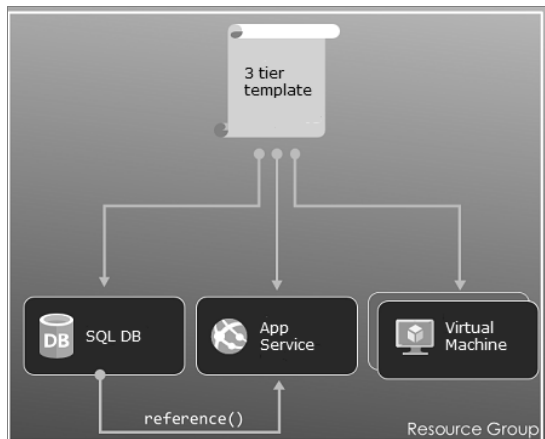


图 11.4-1

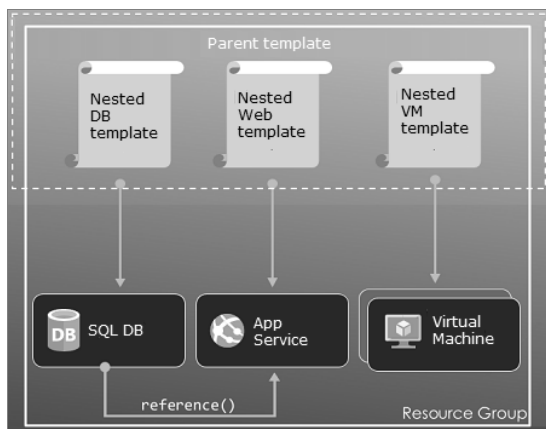


图 11.4-2

如果希望层具有不同的生命周期，可将这三个层部署到不同的资源组。请注意，资源仍可链接到其他资源组中的资源，如图 11.4-3 所示。

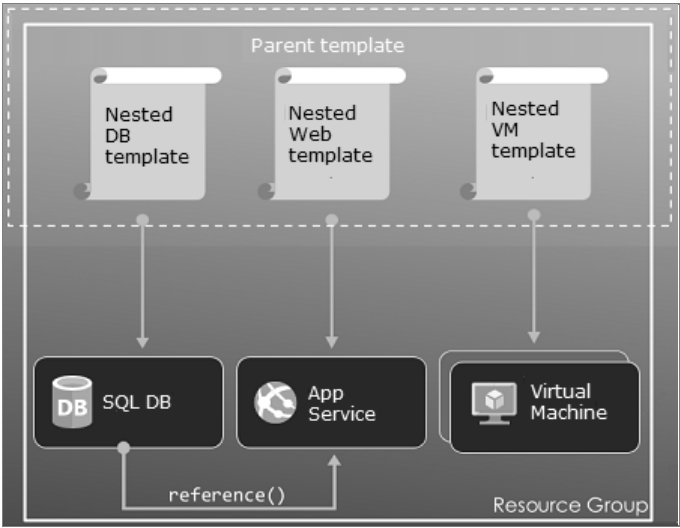


图 11.4-3

Azure Resource Manager 模板格式如下。  
模板的基本语法为 JSON，基本的模板结果包含以下元素：

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "param1": {
      "type": "String",
      "defaultValue": "defaultValue"
    }
  },
  "variables": {
    "var1": "value"
  },
  "resources": [
    {
      "type": "Microsoft.Sql/Servers/databases",
      "name": "[concat('database', param('param1'))]",
      "location": "[location]",
      "properties": {
        "collation": "Latin1_General_CI_AS_KS_WS",
        "compatibility_level": 100,
        "containment": "Contained",
        "edition": "Basic",
        "encrypt_sensitive_data": false,
        "recovery_model": "Full",
        "secondary_log_path": null,
        "secondary_name": null,
        "size": "1GB",
        "status": "Normal"
      }
    }
  ],
  "outputs": {
    "out1": {
      "type": "String",
      "value": "[var('var1')]"
    }
  }
}
```

各元素的解释说明，如表 11.4-1。

表 11.4-1

元素名称	必 选	说 明
\$schema	是	描述模板语言版本的 JSON 架构文件所在的位置。使用前面的示例中显示的 URL。
contentVersion	是	模板的版本（例如 1.0.0.0）。可为此元素提供任意值。使用模板部署资源时，此值可用于确保使用正确的模板。
parameters	否	执行部署以自定义资源部署时提供的值。
variables	否	在模板中用作 JSON 片段以简化模板语言表达式的值。
resources	是	已在资源组中部署或更新的资源类型。
outputs	否	部署后返回的值。

更多详细的模板，可以从 Github 仓库中获取，如下 URL 供参考：  
<https://github.com/Azure/azure-quickstart-templates>  
NOTE：从 GitHub 仓库 "azure-quickstart-templates" 中下载的模板，需要做一些修改

才能适用于 Azure 中国云环境。例如，替换一些终结点，"blob.core.windows.net" 替换成 "blob.core.chinacloudapi.cn"，"cloudapp.azure.com" 替换成 "chinacloudapp.cn"；改掉一些不支持的 VM 映像，还有，改掉一些不支持的 VM 大小。

### 11.4.3 实例展示

#### 实例 1：使用 Azure PowerShell 创建 ARM VM

- (1) 按照 11.1 的部分安装 Azure PowerShell 1.0 以上的版本。
- (2) 使用类似如下命令，登录 ARM 模式：

```
Login-AzureRmAccount -EnvironmentName AzureChinaCloud
```

- (3) 使用类似如下命令，设置需要创建 VM 的订阅：

```
Select-AzureRmSubscription -SubscriptionName Internal-003
```

- (4) 使用类似如下命令，创建资源组：

```
$rgName = "hlmrge1"
$locName = "China East"
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

- (5) 使用类似如下命令，创建存储账户：

```
$stoName = "hlmrstoel"
New-AzureRmStorageAccount -Name $stoName -ResourceGroupName $rgName -Type
Standard_LRS -Location "China East"
```

- (6) 使用类似如下命令，创建虚拟网络：

```
$netName = "hlmrnetel"
$defSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name defSubnet
-AddressPrefix 172.16.0.0/24
$vnnet = New-AzureRmVirtualNetwork -Name $netName -ResourceGroupName $rgName
-Location "China East" -AddressPrefix 172.16.0.0/16 -Subnet $defSubnet
```

- (7) 使用类似如下命令，创建网卡：

```
$nicName = "stonermnic"
$staticIP = "172.16.0.100"
$subnetIndex = 0
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName
-Location $locName -AllocationMethod Dynamic
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName
-Location $locName -SubnetId $vnnet.Subnets[$subnetIndex].Id -PublicIpAddressId
$pip.Id -PrivateIpAddress $staticIP
```

- (8) 使用类似如下命令，设置虚拟机的名字和大小：

```
$vmName = "hlrmvm1"
$vmSize = "Standard_A2"
```

```
$vm = New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize
```

(9) 使用类似如下命令，设置虚拟机的操作系统，网卡，管理员账号等信息：

```
$pubName = "MicrosoftWindowsServer"
$offerName = "WindowsServer"
$skuName = "2012-R2-Datacenter"
$cred = Get-Credential -UserName stone -Message Password
$vm = Set-AzureRmVMOperatingSystem -VM $vm -Windows -ComputerName $vmName -Credential $cred -ProvisionVMAgent -EnableAutoUpdate
$vm = Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"
$vm = Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

(10) 使用类似如下命令，设置系统盘并创建 VM：

```
$diskName = "OSDisk"
$stoAcc = Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $stoName
$osDiskUri = $stoAcc.PrimaryEndpoints.Blob.ToString() + "vhds/" + $vmName + $diskName + ".vhd"
$vm = Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -CreateOption fromImage
New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm
```

实例 2：下载 Github Template，通过使用本地文件的方式部署 ARM VM

### 1. 将 Github Template 文件下载到本地

如图 11.4-4 所示。

<https://github.com/Azure/azure-quickstart-templates/tree/master/101-vm-simple-windows>

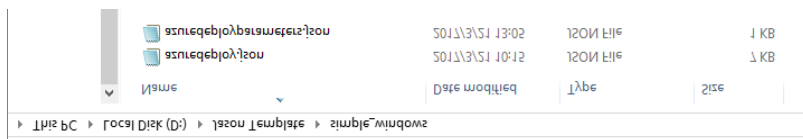


图 11.4-4

### 2. 适当的修改部署文件和参数文件

(1) 修改 azuredeploy.json 文件，如图 11.4-5 和图 11.4-6 所示。

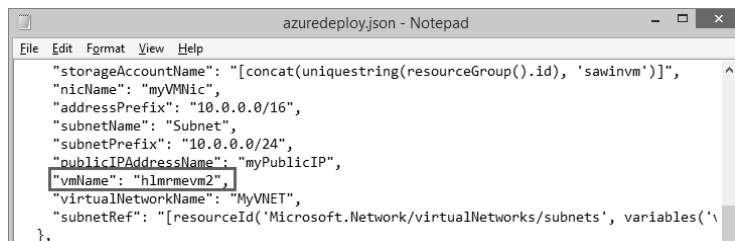


图 11.4-5

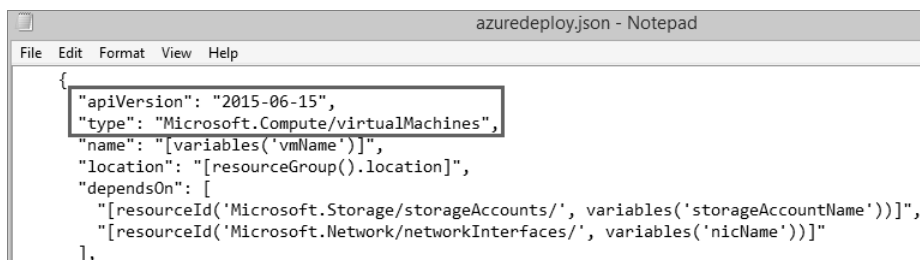


图 11.4-6

(2) 修改 azuredeploy.parameters.json 文件，如图 11.4-7 所示。

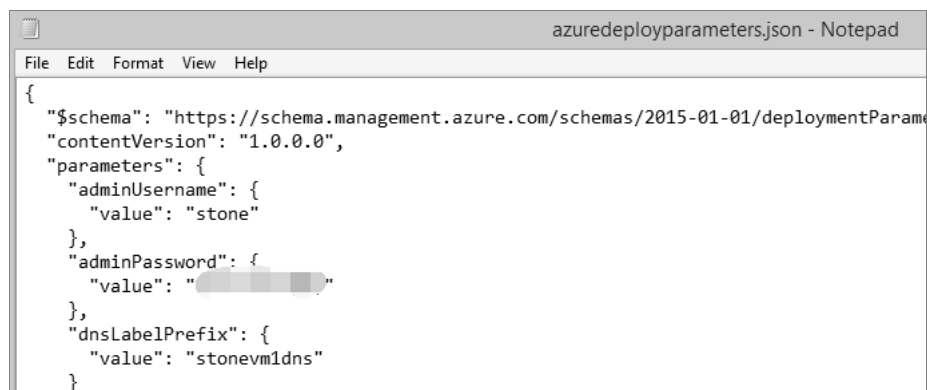


图 11.4-7

Note: 当通过模板部署 VM 失败时，需要仔细阅读报错信息，然后对模板文件进行修改。例如下报错，需要修改“azuredeploy.json”部署文件中“virtualMachines”类型的“apiVersion”，如图 11.4-8 所示。



图 11.4-8

### 3. 应用类似如下命令，使用本地文件通过 Azure Powershell 部署 VM

Note1: 部署模板时，必须指定一个资源组。如果要将 VM 部署到现有资源组，可以直接运行第三小步，然后使用该资源组。

```
$rgName = "hlmrge2"
New-AzureRmResourceGroup -Name $rgName -Location "China North"
New-AzureRmResourceGroupDeployment -Name hlmrge2dep -ResourceGroupName $rgName -TemplateFile "D:\Jason Template\simple_windows\azuredeploy.json" -TemplateParameterFile "D:\Jason Template\simple_windows\azuredeployparameters.json"
```

Note2:

本地部署文件路径: D:\Jason Template\simple\_windows\azuredeploy.json

本地参数文件路径: D:\Jason Template\simple\_windows\azuredeployparameters.json

### 实例 3: 使用外部文件通过 Azure Powershell 部署 ARM VM

Note: 默认 <https://github.com/Azure/azure-quickstart-templates/tree/master/101-vm-simple-windows>, 是不支持被直接编辑的, 如果需要自定义 Json 模板, 需要将该 Github 库 Fork 在自己的 Github 账号下, 从而进行相应的调整。

#### 1. 适当的修改部署文件和参数文件

(1) 修改 azuredeploy.json 文件, 如图 11.4-9 和图 11.4-10 所示。

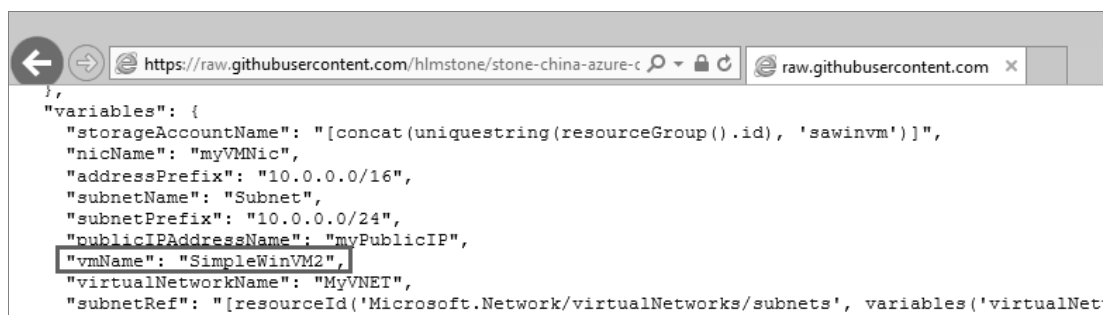


图 11.4-9



图 11.4-10

(2) 修改 azuredeploy.parameters.json 文件, 如图 11.4-11 所示。



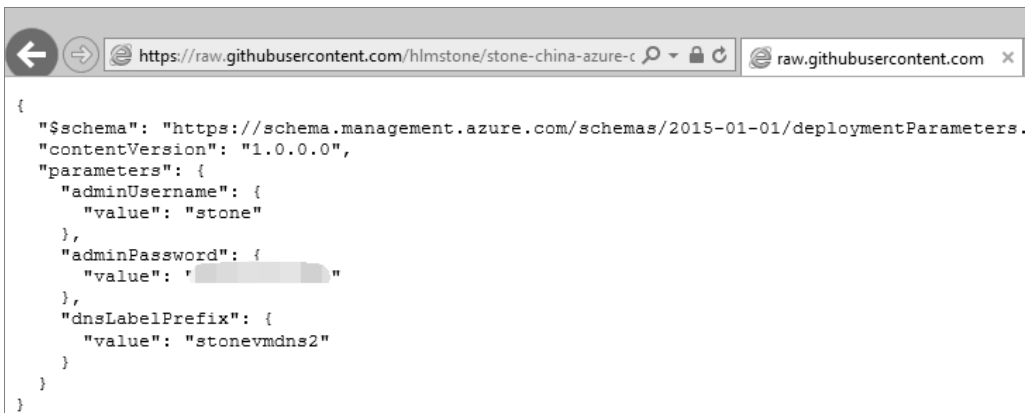


图 11.4-11

## 2. 应用类似如下命令，使用外部文件通过 Azure Powershell 部署 VM

```

$rgName = "hlmrge3"
New-AzureRmResourceGroup -Name $rgName -Location "China East "
New-AzureRmResourceGroupDeployment -Name hlmrge3dep -ResourceGroupName $rgName -TemplateUri "https://raw.githubusercontent.com/hlmstone/stone-china-azure-quickstart-templates/master/101-vm-simple-windows/azuredeploy.json" -TemplateParameterUri "https://raw.githubusercontent.com/hlmstone/stone-china-azure-quickstart-templates/master/101-vm-simple-windows/azuredeploy.parameters.json"

```

### Note:

外部部署文件路径：<https://raw.githubusercontent.com/hlmstone/stone-china-azure-quickstart-templates/master/101-vm-simple-windows/azuredeploy.json>

外部参数文件路径：<https://raw.githubusercontent.com/hlmstone/stone-china-azure-quickstart-templates/master/101-vm-simple-windows/azuredeploy.parameters.json>

## 实例 4：使用 Json Template 通过 Azure Portal 创建 VM

### 1. 适当的修改部署文件和参数文件

(1) 修改 README.md 文件，如图 11.4-12 所示。



图 11.4-12

(2) 修改 azuredeploy.json 文件，如图 11.4-13 和图 11.4-14 所示。



图 11.4-13



图 11.4-14

2. 鼠标右击“Deploy to Azure”，选择“复制快捷方式”，如图 11.4-15 所示

URL: <https://github.com/hlmstone/stone-china-azure-quickstart-templates/tree/master/101-vm-simple-windows>

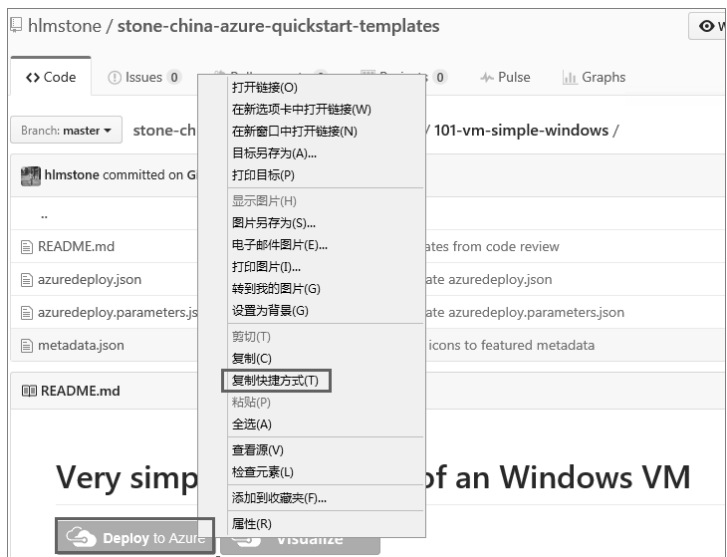


图 11.4-15

修改“portal.azure.com”为“portal.azure.cn”，之后在浏览器中打开，会自动跳转到 Azure New Portal 界面，填写相关参数部署 VM。

(1) <https://portal.azure.cn/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2Fhlmstone%2Fstone-china-azure-quickstart-templates%2Fmaster%2F101-vm-simple-windows%2Fazuredeploy.json>

(2) 如图 11.4-16 所示。

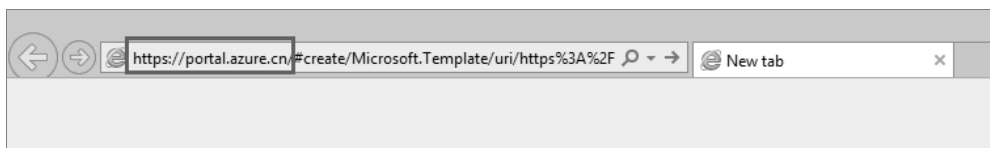


图 11.4-16

(3) 如图 11.4-17 所示。

图 11.4-17

## 第十二章 内容分发网络

CDN（Content Delivery Network，即内容分发网络）是一种通过互联网互相连接的计算机网络系统，提供高性能、可扩展性、及低成本的网络将内容传递给用户。基本原理是广泛使用 Cache 服务器，将这些 Cache 服务器分布到用户访问相对集中的地区或网络中，在用户访问域名网站时，利用全局负载技术将用户的访问指向距离最近且工作正常的 Cache 服务器上，由 Cache 服务器直接响应用户请求。如果 Cache 服务器中没有用户要访问的内容，请求会根据 CDN 的配置进行回源，源服务器会应答相应的请求，返回给用户，同时 Cache 服务器也会根据缓存配置规则决定是否需要在自己服务器上进行缓存。一旦启用 CDN 网络服务后，内容的分布和用户的访问定位全部是自动完成的。其目的是使用户就近取得所需内容，解决由于地域、带宽、运营商接入等 Internet 网络拥挤的状况，最终提高用户访问网站的响应速度。

Azure CDN 通过遍布在中国大陆的众多物理节点上缓存 Azure 平台上的 Storage Blob，Cloud Service 和 WebSites 的静态内容，以及为媒体服务提供流式内容分发提供加速，为开发人员提供一个传送高带宽内容的解决方案。目前 Azure CDN 服务也支持部署在本地或别的服务厂商的源站。同时，Azure CDN 服务是一种融合 CDN 服务，后台整合多家国内优质 CDN 服务，对外提供统一的管理、计费和支持接口：

（1）统一管理：提供统一的管理、监控和运维管理门户，客户无需使用不同 CDN 厂商的自有界面。

（2）统一计费：提供统一计费模型、费率，用多少付多少。

（3）统一支持：提供统一的技术支持接口，统一协调多家 CDN 服务商。

### 12.1 HTTP 加速服务

现有 Azure 用户可以通过 Azure Portal 直接创建基于 HTTP 协议的 CDN 加速服务。

Azure Portal 的 CDN 界面，如图 12.1-1 和图 12.1-2 所示。



图 12.1-1

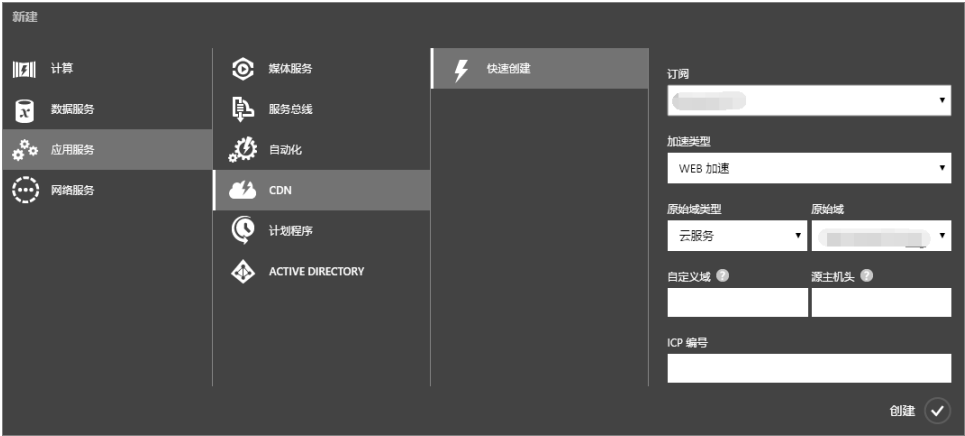


图 12.1-2

### 12.1.1 创建 HTTP 加速服务之前，需要了解的内容

创建 HTTP 加速服务之前，需要预先了解如下概念。

(1) 订阅，如图 12.1-3 所示。



图 12.1-3

如果 Azure 账号下存在多个订阅，需要根据自身需求选择恰当的订阅。

(2) 加速类型，如图 12.1-4 所示。



图 12.1-4

现有 Azure CDN 默认支持 4 种加速类型，分别为：WEB 加速，下载加速，HTTP VOD（视频点播）加速和 HTTP 实时流（视频直播）加速。以下为 4 中加速类型的相关介绍：

WEB 加速，WEB 加速服务是最基本也是应用最广泛的 CDN 加速服务，主要针对 html 文件、CSS、图片、JS、flash 动画等更新频率低的小文件加速。通过将这些小文件缓存到

Azure CDN 的边缘节点，减少源站的访问压力，同时满足用户就近访问网站的需求，提高网站的访问体验，进而带动网站的用户访问量。**WEB 加速 CDN** 节点适用于面向访问量较大的大中小企业门户类网站。如政府机构网站，企业门户网站等。

下载加速，下载加速主要针对 20MB 以上的大文件下载，例如软件安装包、游戏客户端、应用程序、影音等大文件的下载分发。Azure CDN 将文件缓存到 CDN 边缘节点，缓解源站下载的带宽压力，提高用户下载体验。下载加速适用于操作系统固件升级，游戏客户端，手机 APP 更新，应用程序下载等用户场景。

**HTTP VOD（视频点播）加速**，VOD 视频点播加速服务主要针对在线音视频点播提供加速服务。随着网路视频媒体服务的兴起，越来越多的用户选择使用网络平台收听观看各种音视频。加之国内网络环境的限制，对音视频内容的最终分发提出了非常高的要求。Azure CDN 将音频、视频等流媒体内容分发缓存到 CDN 边缘节点，将用户请求指向最优节点，减少源站服务器的负载，节省带宽资源，给用户提供高速、流畅、高质量的在线视频体验。Azure CDN VOD 视频点播加速支持 Azure 内置的媒体服务。VOD 视频点播加速适用于各类在线音视频点播网站，如媒体类视频网站，在线教育网站，移动端 APP 客户端等。

**HTTP 实时流（视频直播）加速**，流媒体直播加速服务主要针对在线视音频播提供加速服务。网络直播服务快速、实时的特性备受广大用户青睐。直播的实时性导致海量用户并发访问，给源站和带宽资源带来巨大压力，同时受国内网络跨地域跨运营商的限制，对高质量、快速的流媒体直播提出了较高的要求。Azure CDN 流媒体加速服务通过实时采集源站视频流并分发到距离用户最近的 CDN 边缘节点，通过智能缓存和调度策略，为用户提供计算最优节点，减少链路传输造成的延迟和带宽压力，且按使用量付费，给用户提提供高速、流畅、高质量的直播观看体验。Azure CDN 流媒体直播加速主要基于 HTTP Live Streaming(HLS)协议，且支持 Azure 内置的媒体服务。流媒体直播加速适用于各类流媒体直播网站，如网络电视直播，体育赛事，盛典赛事直播等。

（3）原始源类型，如图 12.1-5 所示。

现有 Azure CDN 默认支持 5 种原始域类型，分别为：云服务，存储账户，WEB 应用，媒体服务（media service）和自定义原始域。需要选择适合自身应用服务的原始域类型，以便下一步中选择正确的原始域。

（4）原始域，如图 12.1-6 所示。

原始域又名源站，是 CDN 所缓存内容的原始位置，一般其中会部署相应的应用服务或者可以访问到应用服务的域名地址，请选择正确的原始域名配置 CDN 服务。



图 12.1-5



图 12.1-6

(5) 自定义域名，如图 12.1-7 所示。

自定义域名又名加速域名，是用于访问 CDN 缓存内容的 URL。为预先准备的供终端客户访问的域名，如果没有需要到相关域名供应商处注册或购买。该自定义域名可支持通配符域名，如：\*.yourcompany.com。

(6) 源主机头，如图 12.1-8 所示。



图 12.1-7

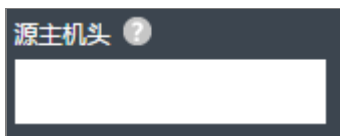


图 12.1-8

源主机头为源站所接受的回源访问 Host Header。当输入完“自定义域”之后，系统会根据所选择的“原始域类型”来自动填充一个默认值。具体的规则是，如果源站是在 Azure 上的话，默认值就是相应的源站地址。如果源站不在 Azure 上，默认值是输入的“自定义域名”，当然也可以根据自己源站的实际配置情况来修改。但请确保使用输入的域名能访问您的源服务器。

(7) ICP 编号，如图 12.1-9 所示。



图 12.1-9

该处需要填写与自定义域名相对应的 ICP 备案号(格式如:京 ICP 备 XXXXXXXXX 号-X)

## 12.1.2 通过 Azure Portal 创建一个 Http CDN 加速服务

通过 Azure Portal 创建一个 CDN 服务的步骤如下。

预先准备环境：

原始域（源站）：hlmcloud.chinacloudapp.cn

自定义域名（加速域名）：www.2dream.com.cn

ICP 编号：辽 ICP 备 13000266 号。

原始域可以被成功访问，如图 12.1-10 所示。

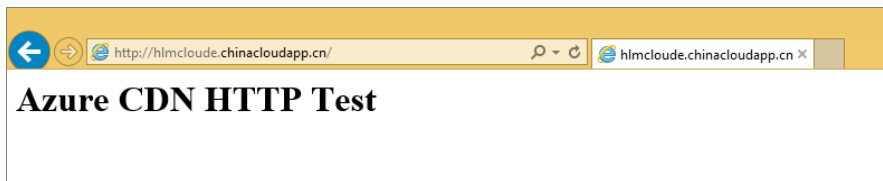


图 12.1-10

(1) 按照上述概念介绍填写 CDN 配置界面的相关参数，如图 12.1-11 所示。

快速创建

订阅

加速类型

WEB 加速

原始域类型

云服务

原始域

hlmcloud.chinacloud

自定义域 ?

www.2dream.com.cn

源主机头 ?

hlmcloud.chinacloudap

ICP 编号

辽ICP备13000266号

创建

图 12.1-11

(2) 单击“创建”按钮以创建新的 CDN 终结点。终结点创建后将出现在订阅终结点的列表中。列表视图显示了用于访问缓存内容的自定义域以及原始域，如图 12.1-12、图 12.1-13、图 12.1-14 所示。

✓ 已成功创建 CDN 终结点。

确定

+ 新建

管理

禁用

删除

1 1 ?

图 12.1-12

cdn						
名称	状态	订阅	自定义域	原始域	ICP 编号	终结点类型
CDN_www.2dream.com.cn →	CName 非活动		www.2dream.com.cn	http://hlmcloud...	辽ICP备13000266号	付费版

图 12.1-13



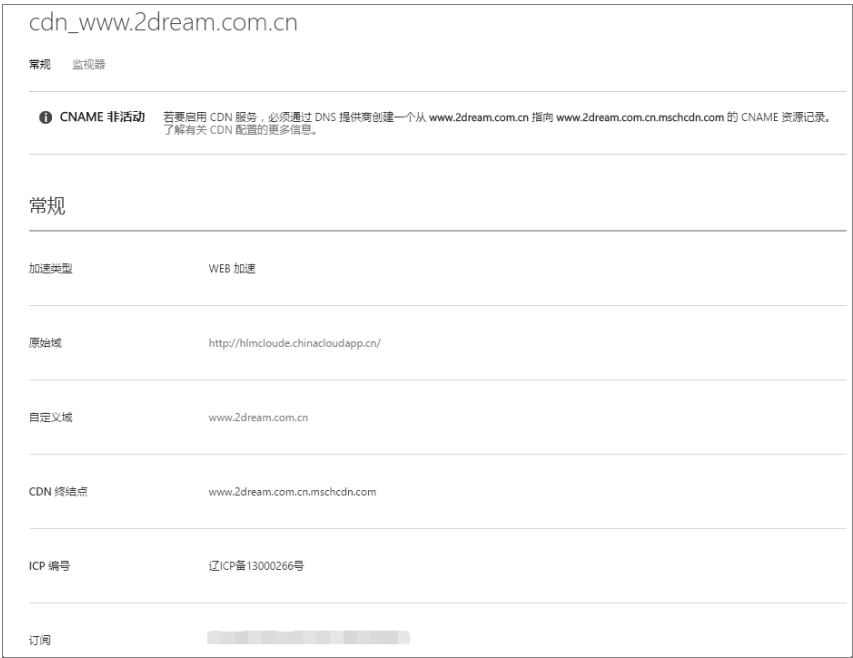


图 12.1-14

(3) 需要根据提示信息在自定义域名的域名空间中填写对应的 CName 映射信息，这样才能真正启用 CDN 服务，如图 12.1-15、图 12.1-16、图 12.1-17、图 12.1-18 所示。



图 12.1-15



图 12.1-16



图 12.1-17



图 12.1-18

等待片刻后，发现自定义域名：`www.2dream.com.cn`，可以成功访问。至此 Azure CDN HTTP 加速服务已经配置完成，如图 12.1-19 所示。



图 12.1-19

## 12.2 HTTPS 加速服务

现有 Azure CDN 支持对 HTTPS 协议的加速，且仅对 Azure 付费用户开放。默认 Azure 账号订阅下并没有开通可以创建 CDN HTTPS 加速服务的权限。

通过 Azure Portal 创建一个 HTTPS CDN 加速服务的步骤如下。

(1) 开通申请。

联系 Azure 技术支持团队进行开通申请，同时提供需要开通 HTTPS 加速服务的 Azure 订阅 ID。

(2) 目前 HTTPS 的加速服务可支持在 Azure Portal 上自助式创建。

按照步骤 1 开通指定订阅 CDN HTTPS 的权限后，可以通过 Azure Portal 创建 HTTPS 的加速服务，在 Azure 管理门户完成 HTTPS 加速类型的创建之后，可以通过单击 CDN 配置界面的“管理”按钮，跳转到 Azure CDN 高级管理门户进行后续的详细参数配置，如图 12.2-1 和图 12.2-2 所示。



图 12.2-1



图 12.2-2

(3) SSL 证书申请和配置。

收到相关配置请求后，Azure CDN 后台会代为申请 SSL 证书，此证书申请配置时间大约需要五个工作日，同时在申请过程中，需要配合证书签发机构确认域名所有权。具体操作方式，可参阅本书的伴侣网站以获得最新信息。

有关证书的详细说明见下：

SSL 证书类型为 SAN 多域名证书（SAN/UCC SSL）：

SAN 证书-Subject Alternative Name certificates，又称为 UCC 证书- Unified Communication Certificates。SAN SSL 证书允许在同一张证书中，添加多个需要保护的“域名”或“服务器”名。这种功能提供了非常大的使用弹性，它可以创建一张易于使用和安装却又比通配符 SSL 证书更安全，完全适合服务器安全需求的 SSL 证书。该 SSL 证书由 Azure CDN 代为申请，安装和维护。证书签发机构为：<https://www.digicert.com/>。

(4) 全部配置完成后，可以像创建 Azure CDN HTTP 加速类型服务一样，做最后的 CNAME 记录设置。之后，就可以通过统一的 Azure CDN 高级管理门户查看或配置相应的 CDN 服务功能。

## 12.3 缓存规则

Azure CDN 缓存规则，需要在高级管理门户的“域名管理”项中设置，如图 12.3-1 所示。



图 12.3-1

12.3.1 Azure CDN 默认缓存规则

默认的缓存规则配置视图，如图 12.3-2 所示。

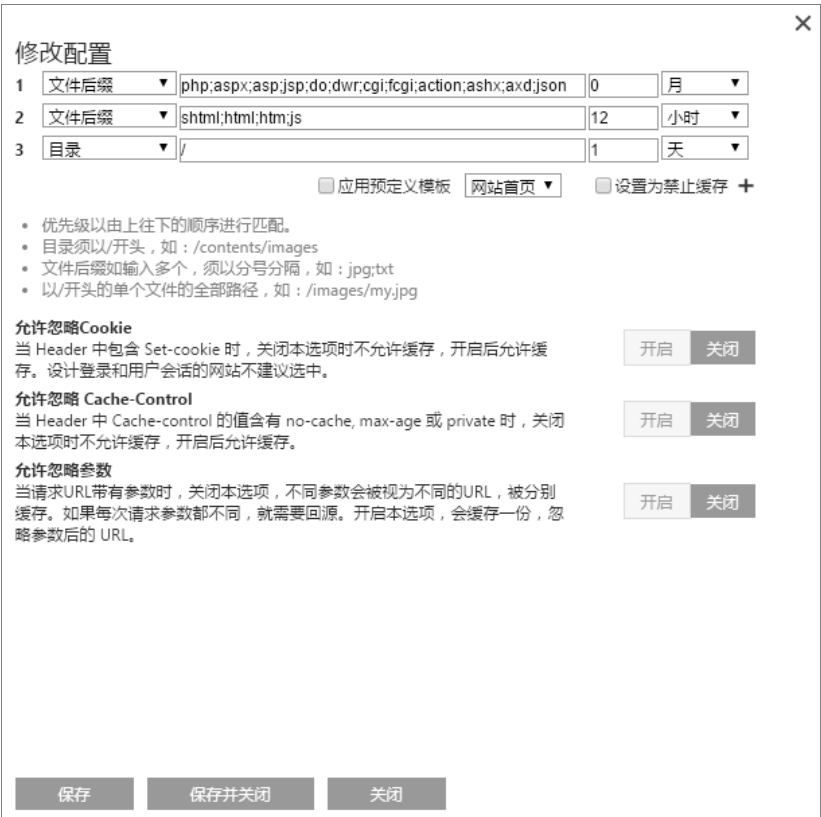


图 12.3-2

**Note:** 系统会根据缓存规则设置默认规则。可以根据具体环境需求加以调整，用户规则优先匹配，如果用户规则未命中，则逐条执行系统默认缓存规则。

## 12.3.2 Azure CDN 缓存规则配置

### 缓存规则配置

单击“配置缓存规则”后，可以根据需求设置对域名的缓存规则，包括：

#### 1. 根据目录进行配置

目录必须以 "/" 开头，比如："/pic"，"/doc"，"/htdocs/data" 等等。后台会匹配指定目录下的所有文件，包括子目录。

#### 2. 根据文件后缀配置

常用文件后缀名，比如："jpg"，"png"，"gif"，"txt"，"m4v"，"mp3" 等等。后台会匹配所有文件夹下指定的文件后缀。

#### 3. 根据全路径配置

用来指定一个文件，必须以 "/" 开头。比如："/sites/doc/example.doc"。注意：如果用户填的全路径是 "/"，则它匹配首页。

**Note:**

- (1) 用户填写配置规则时，字符串中不要包含“{”，“}”，“（”，“）”，“[”，“]”，“.”，“?”，“\*”，“\”，“^”，“\$”等特殊字符。
- (2) 时间填为 0 表示禁止缓存。

### 缓存配置顺序

系统根据配置顺序逐条匹配，最先配置的规则具有最高优先级。规则被匹配后，其后的规则不再被匹配。

### 预定义模板

可以通过“应用预定义模板”快速创建缓存配置规则。下图列出了选中“应用预定义模板”后，选择“常见文件”后创建的一个规则。用户可以根据需求对自动创建的规则进行修改，如图 12.3-3 所示。

修改配置			
1	文件后缀	php;aspx;aspjsp;do;dwr;cgi;fcgi;action;ashx;axd,json	0 月
2	文件后缀	shtml;html;htm.js	12 小时
3	目录	/	1 天
<input checked="" type="checkbox"/> 应用预定义模板    常见文件 <input type="checkbox"/> 设置为禁止缓存 +			

图 12.3-3

### 禁止缓存设置

勾选“设置为禁止缓存”，则该加速域名将不会被缓存，如图 12.3-4 所示。



图 12.3-4

## 12.4 日志查看

Azure CDN 日志，需要在高级管理门户的“日志下载”项中设置，如图 12.4-1 所示。



图 12.4-1

Note：现有 Azure CDN 日志不支持在线查看，只能将日志下载后进行查看

### 12.4.1 Azure CDN 日志下载

(1) 日志下载需要用户提供一个 Azure Storage Account 用以存放 CDN 日志，该 Storage Account 需要通过单击“下载设置”进行设置，如图 12.4-2 所示。



图 12.4-2

(2) 填写指定的存储账号信息（该信息可以通过“存储空间”—“管理访问密钥”出进行获取），并进行有效性验证，如图 12.4-3 所示。

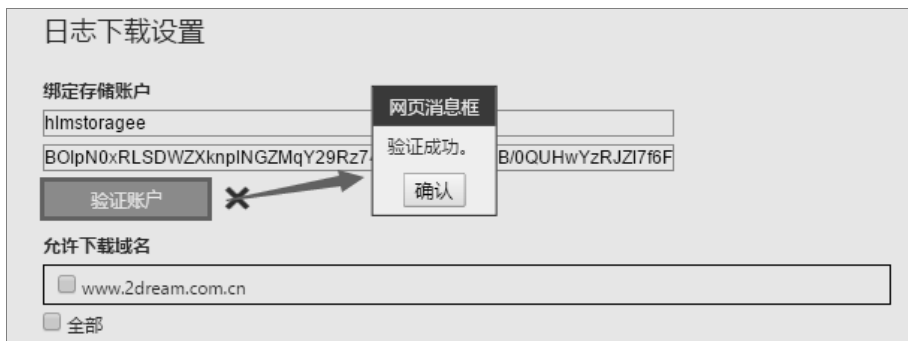


图 12.4-3

(3) 勾选需要下载的 CDN 日志对应的自定义域名，之后单击“保存”按钮，指定存储账号下会自动创建一个名称为“cdn-access-logs”的容器，同时日志会自动下载到该容器下，如图 12.4-4、图 12.4-5 所示。



图 12.4-4



名称	URL
bootdiagnosticsf120779f-df94-4b3d-abd3-002638250	https://hlmstorageee.blob.core.chinacloudapi.cn/bootdiagnosticsf120779f-df94-4b3d-abd3-00263825063e
bootdiagnosticsf66b93dd-950e-449f-9c37-ca154f5e6e	https://hlmstorageee.blob.core.chinacloudapi.cn/bootdiagnosticsf66b93dd-950e-449f-9c37-ca154f5e6ecc
cdn-access-logs	https://hlmstorageee.blob.core.chinacloudapi.cn/cdn-access-logs
vhds	https://hlmstorageee.blob.core.chinacloudapi.cn/vhds
vmdepot-images	https://hlmstorageee.blob.core.chinacloudapi.cn/vmdepot-images

图 12.4-5

12.4.2 Azure CDN 日志命名格式

Azure CDN 日志有统一的命名格式规范，日志格式详情见下：

日志以 blob 的形式存放在名为 "cdn-access-logs" 的容器中。每个 blob 是一个 GZip 压缩后的 CSV 文件。其中每一栏的含义如下：

c-ip: 客户端 IP 地址

timestamp: 访问时间

cs-method: HTTP 请求动作，如 GET/HEAD 等。

cs-uri-stem: 请求的 URI

http-ver: HTTP 协议版本

sc-status: HTTP 状态码

sc-bytes: 服务器向客户端传送的字节数

c-referer: 客户端 Referer URI

c-user-agent: 客户端 User Agent 标识

rs-duration(ms): 完成请求花费的时间（单位毫秒）。

hit-miss: CDN 缓存命中、丢失标识。

s-ip: 生成日志的 CDN 边缘节点 IP 地址。

Note:

（1） 如果 CDN 日志中未包括栏目内容，则相应记录标记为“-”，比如“c-referer”记录。此外，取决于边缘节点的日志配置，“rs-duration”、“hit-miss”、“s-ip”等记录也有可能为空。

（2） 用户也可以删除存储账号以取消日志下载。

12.5 FAQ

1. 刚刚创建的 CDN 终结点是否可以立即使用？

答：刚刚创建的 CDN 终结点并不能立即使用，首先 CDN 后台需要审核所提供的自定义域名和 ICP 编号是否匹配、有效。这个过程需要最多一个工作日的时间来完成。如果 ICP



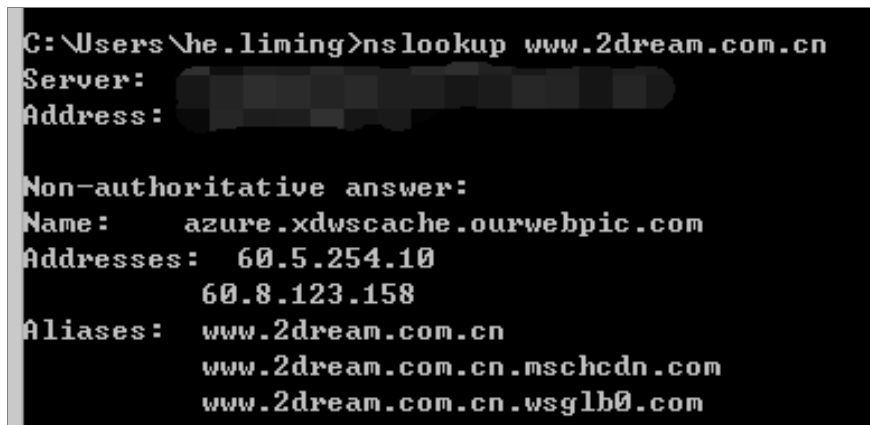
审核通过，CDN 服务最多需要 60 分钟时间进行注册以便通过 CDN 网络传播。与此同时，您还需要按照界面上的提示信息配置 CNAME 映射信息，这样才可以最终通过自定义域名访问 CDN 缓存内容。如果 ICP 审核没有通过，您需要删除之前创建的这个 CDN 终结点，然后使用正确的自定义域名和 ICP 编号重新创建。

## 2. 使用一级域名的不同二级域名创建多条 CDN 链路时，每个二级域名是否需要备案？

答：同一个一级域名的不同二级域名不需要备案。例如：test.cn 已经备案，那么 www.test.cn; file.test.cn; image.test.cn 则不需要备案，创建 CDN 加速节点时仅需提供 test.cn 的备案号即可。

## 3. 如何确认配置的 CNAME 记录已经生效？

答：各地 DNS 的生效时间不一致，取决于自定义域名对应的原有记录的生效时间（TTL 时间）。当 ping（或者 dig）自定义域名，给出的解析不再是源站 IP，说明已经生效了。同时，nslookup 会解析出类似如下信息，如图 12.5-1 所示。



```
C:\Users\he.liming>nslookup www.2dream.com.cn
Server: 
Address: 

Non-authoritative answer:
Name:      azure.xdwscache.ourwebpic.com
Addresses: 60.5.254.10
           60.8.123.158
Aliases:   www.2dream.com.cn
           www.2dream.com.cn.mschedn.com
           www.2dream.com.cn.wsglb0.com
```

图 12.5-1

## 4. CDN 设置成功后，如何保证缓存内容和源站的同步？

答：设置缓存规则，针对不同的内容设置不同的缓存刷新规则，对更新频繁的内容，可以设置较短的缓存时间；对于不经常更新的内容，可以设置较长的缓存时间，从而减小源站压力。若设置的缓存刷新周期未到，但是有新内容发布或者删除部分内容，可以使用 Azure CDN 管理平台提供的缓存刷新功能，进行手动强行刷新。另外，如果只更新某个文件，建议使用文件刷新对更新的文件进行刷新。目录刷新会针对目录下所有文件进行刷新，生效时间比较慢。

## 5. 回源流量应该小于 CDN 流量，为什么有时候会出现回源流量大于 CDN 流量？

答：正确情况下回源流量小于或等于 CDN 流量，特殊情况也可能出现回源流量大于 CDN 流量，比如：如果访问者发起一个请求，请求一个比较大的文件。比如 150MB，如果节点没有缓存的情况，CDN 节点就会去源站获取，这么大文件必然需要点时间，然而此时，访问者又不继续等了于是就断开连接。这样，CDN 节点还是会去把 150MB 文件全都

拿过来。此时发现访问者已经不要它了，它就没法再返回了，于是，回源的流量就有 150MB，而 CDN 流量为空。

## 6. 为什么有些 URL 没有被缓存？

答：

(1) 源站的该 URL 响应 Header 里含有以下信息：

Set-Cookie（且缓存规则里并未勾选忽略 Set-Cookie 选项）。注：Set-Cookie 在用于用户登录和身份识别时是不能勾选忽略 Set-Cookie 选项的，否则可能引起功能性问题。

Cache-Control: no-store/no-cache/private（且缓存规则里并未勾选忽略 Cache-Control 选项）。

Expires 的时间是过去的某个时间，Expires 指定了缓存到期时间点，如果是过去时间，则将导致无法缓存。

Max-age 的值很小，Max-age 指定了缓存时间长度，单位为秒，如果太小，如小于两位数，那么很快就会过期，导致无法缓存。

(2) 缓存规则里没有配置或配置错误，URL 无法命中任何一个缓冲规则，例如，有用户不小心录入以下规则：" [任意字符](.gif|.jpg|.bmp)(.gif|.jpg|.bmp) "，那么即使是图片类型也无法命中规则，因为扩展名重复。

(3) 部分节点暂时还没有用户访问该 URL，需要有访问之后才会缓存。

7. 如果出现 CDN 加速域名无法访问的情况，应该搜集哪些信息反馈给 Azure 技术支持团队？

答：

(1) 问题 URL（最好以可复制的文本格式），访问问题 URL 时的问题截图说明。

(2) nslookup 解析自定义域名的输出信息，如图 12.5-2 所示。

```
C:\Users\he.liming>nslookup www.2dream.com.cn
Server: 
Address: 

Non-authoritative answer:
Name:      azure.xdwscache.ourwebpic.com
Addresses: 60.5.254.10
           60.8.123.158
Aliases:   www.2dream.com.cn
           www.2dream.com.cn.mschedn.com
           www.2dream.com.cn.wsg1b0.com
```

图 12.5-2

(3) 终端客户的出口公网 IP。

(4) 发生问题的时间点。

(5) Azure Portal 界面下 CDN 的配置信息，如图 12.5-3 所示。



图 12.5-3

8. 如果自定义域名为泛域名时，可以配置缓存规则？支持缓存刷新？CDN 的内容预取功能是否可以应用于泛域名？

答：

(1) 可以给泛域名配置缓存规则，创建了泛域名后，缓存规则的设定就是在该泛域名下。泛域名主要是针对多个域名配置相同来使用的，简化了创建的步骤。它可以与真正的域名同时创建，真正的域名的配置会优先匹配。例如，如果有个 a.test.cn 的规则是不一样的，客户可以新建一个 a.test.cn 的 endpoint，并在这里面创建缓存规则，这里的配置会优先于 \*.test.cn 的配置。

(2) 如果自定义域名为泛域名，提交缓存刷新的时候，刷新 URL 必须指定子域名。比如：自定义域名为 \*.test.cn，如果要刷新 file.test.cn 下的内容，则应指定子域名 file.test.cn 进行缓存刷新。

(3) 内容预取功能应用的对象必须是子域名，并且是能够正常访问的 URL（状态码是 200 的）。

9. CDN 日志的生成是否是实时的，具体的生产机制是什么？日志保留的期限为多少？

答：

(1) CDN 日志记录的生成不是实时的，根据 CDN 的供应商不同，生成时间的选择方

式有所不同。但主要有以下两种方式：

- a. 按小时的是每小时生成一次，将距现在 11 到 12 小时之间的日志保存下来。如 14 点会生成当天 2 点到 3 点之间的日志；2 点生成前一天 14 点到 15 点的日志。
- b. 按天的是每天中午 12 点左右生成前一天 0 点到 24 点的日志。

默认情况下是按天的方式记录 CDN 日志的，您可以考虑开通按小时记录 CDN 日志的方式，这样获取到最新日志的时间会更快一些。

（2）在 Azure CDN 中设置了日志下载后，日志会定期下载到用户配置的存储账户中，CDN 不会删除已下载的日志。用户可视情况处理已下载的日志文件。

若客户没有配置日志下载，又需要日志记录，根据供应商不同，我们能提供的日志的时间范围也不同。一般，一个月内的日志可以提供。

10. 创建 CDN 终结点后，域名空间中添加 CName 记录之前，如何验证创建的 CDN 终结点是有效的？

答：可以通过两种方法验证该信息。

（1）在 CDN 高级管理门户中，通过“服务检查”项对加速域名进行检测，如图 12.5-4 和图 12.5-5 所示。



图 12.5-4



图 12.5-5

如上视图，选择需要检查的域名后，提供一个源站可以访问的资源，然后单击“检查”项。

- a. 源站正常，表明提供的资源可以访问；
- b. CDN 服务部署完成，表明该域名对应的 CDN 服务已经部署；
- c. CDN 缓存正常，表明通过源站访问的内容和通过 CDN 访问的内容一致。

(2) 通过绑定本地机器的 hosts（默认路径：C:\Windows\System32\drivers\etc\hosts）文件测试 CDN 终结点的可用性。

a. 绑定 hosts 文件前访问下源站域名，并使用 F12 开发工具查看下“Response headers”项，如图 12.5-6 所示。



图 12.5-6

b. 访问生成的 CDN 终结点（www.2dream.com.cn.mschedn），发现该终结点值不支持访问的，如图 12.5-7 所示。

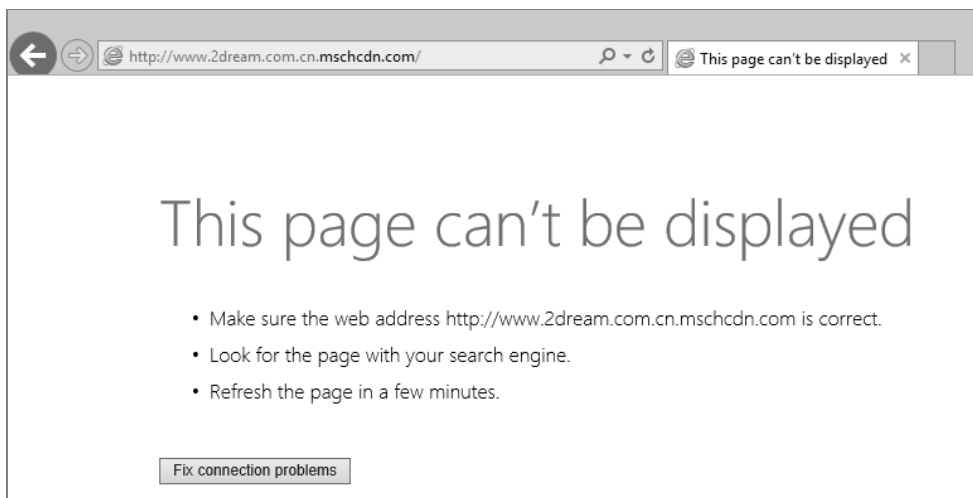


图 12.5-7

c. 使用本地机器的 CMD ping 生产的 CDN 终结点，解析出该终结点对应的 IP 地址，如图 12.5-8 所示。

```
C:\Users\he.liming>ping www.2dream.com.cn.mschedn.com

Pinging azure.xdwscache.ourwebpic.com [122.225.28.145] with 32 bytes of data:
Reply from 122.225.28.145: bytes=32 time=26ms TTL=53
Reply from 122.225.28.145: bytes=32 time=26ms TTL=53
Reply from 122.225.28.145: bytes=32 time=26ms TTL=53
Reply from 122.225.28.145: bytes=32 time=25ms TTL=53

Ping statistics for 122.225.28.145:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 25ms, Maximum = 26ms, Average = 25ms
```

图 12.5-8

d. 编辑本地机器的 hosts 文件，将上一步中解析出的 IP 地址与加速域名的对应关系添加进去，如图 12.5-9 和图 12.5-10 所示。

This PC > Local Disk (C:) > Windows > System32 > drivers > etc				
Name	Date modified	Type	Size	
hosts	2017/2/12 12:25	File	1 KB	
lmhosts.sam	2013/8/22 23:35	SAM File	4 KB	
networks	2013/8/22 21:25	File	1 KB	
protocol	2013/8/22 21:25	File	2 KB	
services	2013/8/22 21:25	File	18 KB	

图 12.5-9

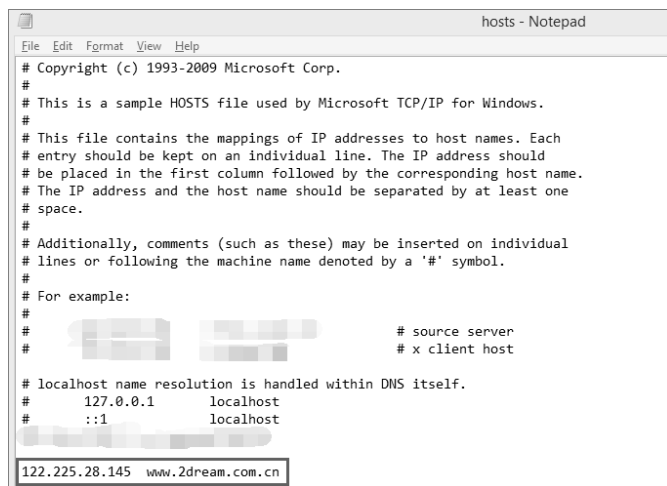


图 12.5-10

e. 再次访问访问下源站域名，并使用 F12 开发工具查看下“Response headers”项，示例如下，发现该项中多出了“X-Cache”，“X-Via”两项，这样的话说明 CDN 终结点的配置已经成功生效，如图 12.5-11 所示。

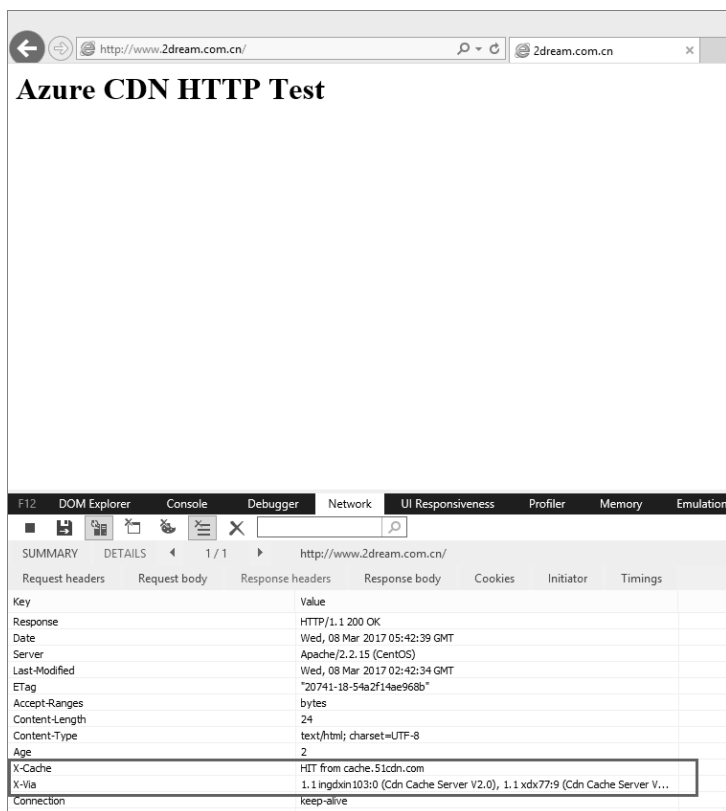


图 12.5-11

## 第十三章 Azure 活动目录

本章详细介绍了 Azure AD 服务的作用，与本地 AD 服务的区别和联系，并结合实际案例分析了一些与 Azure AD 常见的报错信息的分析方法以及原因。此外，本章中针对如何在 Azure AD 中使用自定义域名提供了详细的操作方法，并针对 Azure AD 与本地 AD 用户同步的各种错误给予了分析解答。

### 13.1 Azure 活动目录简介

#### 13.1.1 什么是 Azure 活动目录

Azure Active Directory (Azure AD) 是 Microsoft 提供的基于多租户云的目录和标识管理服务。

对于 IT 管理员而言，Azure AD 提供经济实惠、易于使用的解决方案，使员工和业务合作伙伴能够使用单一登录 (SSO) 功能来访问云中 SaaS 应用程序，例如 Office365。

对于应用程序开发人员而言，Azure AD 可让你专注于构建应用程序，快速方便地集成数百万个全球各地组织所用的一流标识管理解决方案。

Azure AD 还包含整套标识管理功能，例如多重身份验证、设备注册、自助密码管理、自助组管理、特权账户管理、基于角色的访问控制、应用程序使用情况监视、多样化审核以及安全监视和警报。这些功能可以帮助保护基于云的应用程序的安全，简化 IT 流程，削减成本，以及确保实现公司的合规目标。

此外，Azure AD 能与现有的 Windows Server Active Directory 集成，使组织能够运用现有的本地标识管理系统投资来管理对基于云的 SaaS 应用程序的访问。

#### 13.1.2 Azure AD 与本地 Active Directory 域服务 (AD DS) 的不同

Azure Active Directory (Azure AD) 和本地 Active Directory (Active Directory 域服务或 AD DS) 都是存储目录数据和管理用户和资源之间通信 (包括用户登录过程、身份验证和目录搜索) 的系统。

AD DS 是 Windows Server 上的服务器角色，这意味着可将它部署在物理计算机或虚拟机上。它具有基于 X.500 的层次结构。它使用 DNS 查找对象，可使用 LDAP 与它交互，并且它主要使用 Kerberos 进行身份验证。除了将计算机加入域之外，Active Directory 还启用组织单位 (OU) 和组策略对象 (GPO)，并在域之间创建信任。

Azure AD 是多用户公共目录服务，这意味着可在 Azure AD 内为云服务器和应用程序 (如 Office 365) 创建租户。在平面结构中创建用户和组，无需 OU 或 GPO 的。通过



协议（例如 SAML、WS 联合身份验证和 OAuth）执行身份验证。可以查询 Azure AD，但必须使用称为 AD 图形 API 的 REST API 而不是使用 LDAP。这些操作均通过 HTTP 和 HTTPS 运作。

## 13.2 关于 Azure AD 相关案例分析

（1）用户既有 Azure 订阅，又有 O365 的订阅，如何实现两个 AD 用户自动同步

如果用户都已经分别拥有了 Azure 的 Domain 和 O365 的 Domain，就无法自动同步两个 AD 的用户信息了。

如果用户想实现 Azure 与 O365 使用同一个 AD。目前，有以下两种方式选择：

- 如果用户先拥有 Azure 的 Domain。使用该 Domain 申请 O365 服务即可。
- 如果用户先拥有 O365 的 Domain。使用该 Domain 申请 Azure 服务即可。

这样就保证两个服务使用相同的 AD，用户自动保持一致了。

（2）在 Azure 活动目录中添加一个用户，登录 Azure 经典管理门户时，报“我们无法找到你作为服务管理员或协同管理员的任何 Azure 订阅”的提示，请参考图 13.2-1。



图 13.2-1

该问题是由于，虽然在 Azure AD 中已经添加该用户，但是还未给用户绑定订阅信息，具体操作步骤如下：

（1）登录 Azure 经典管理门户后，跳转到“设置”选项卡，选择“管理员”标签，单击“添加”按钮，请参考图 13.2-2。



图 13.2-2

(2) 请在以下位置添加之前在 Azure AD 中新用户名称，并且勾选您要绑定的订阅信息。目前中国区 Azure 添加协调管理员时，不支持 Microsoft Live ID（如 Outlook 或者 Hotmail 邮箱地址）请参考图 13.2-3。



图 13.2-3

完成以上操作后，相应用户就可以正常登录 Azure 经典管理门户了，登录之后，根据之前 Azure AD 中组织角色，可能某些用户无法操作 Azure AD。

如果之前创建在 Azure AD 的用户的组织角色为“用户”请参考图 13.2-4，该用户登录经典管理门户时，没有对 AD 操作的权限。提示信息如下，请参考图 13.2-5。



图 13.2-4



图 13.2-5

如果之前创建在 Azure AD 的用户的组织角色为“全局管理员”，该用户登录经典管理门户时，有 AD 操作的权限。

### 13.2.3 Azure 的用户后缀是否可以调整为用户自定义域名

(1) 在 Azure AD 中添加一个自己的域名（该域名需在互联网上可以解析），单击以下截图中“添加”按钮，请参考图 13.2-6。



图 13.2-6

(2) 输入您组织拥有的域名，单击“添加”按钮，请参考图 13.2-7。



添加域

## 指定域名

输入你的组织拥有的域名。 ?

域名

azurechina.com

☐ 我计划配置此域为使用本地 Active Directory 进行单一登录。 ?

添加

图 13.2-7

(3) 下一步请将您看到如下信息，请参考图 13.2-8，在域名注册结构中添加一条 TXT 记录，其中“目标地址或指向地址”信息，请不要做任何修改，该信息有误，会导致验证失败。



添加域

## 验证 azurechina.com

转到你的域名注册机构网站并更新 azurechina.com 的 DNS 设置。

[如何向常见域名注册机构添加 DNS 记录的说明](#)

为 azurechina.com 添加你的域名注册机构支持的记录类型。

记录类型: TXT 记录

别名或主机名: @

目标地址或指向地址: MS=ms73518831

TTL: 1 小时

验证

1

← ✓

图 13.2-8

(4) 验证通过后，会在 Azure AD 中看到您自定义的域名的状态为“已验证”，请参考图 13.2-9。



图 13.2-9

(5) 后续您在创建新用户时，就可以选择自己的自定义域名了，请参考图 13.2-10。



图 13.2-10

### 13.2.4 执行 Azure AD 目录同步时，遇到 AttributeVauleMustBeUnique 报错

用户的需求为：Azure AD 中存在与本地 AD 中同名用户，同步完成后，是否属性可以更改为“本地”。

按照用户本地环境进行如下测试。

## 1. 本地域控

Windows server 2012 R2 作为域控 azuretest.com，并且添加一个 UPN 后缀名为 azuretestbat.com，请参考图 13.2-11。

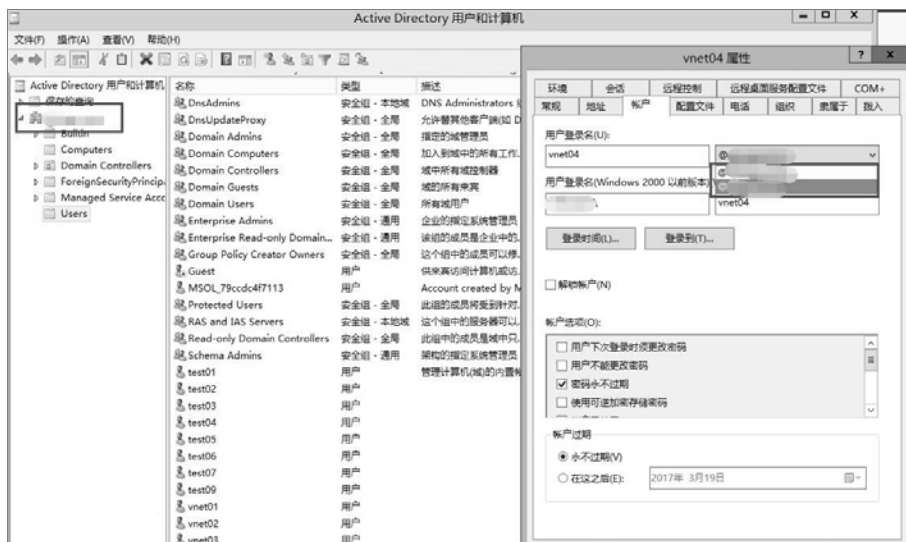


图 13.2-11

## 2. Azure 端环境

azuretestbat.com 已经添加到 Azure AD 自定义域中。目前 azure 端的主域为 \*\*\*\*\*.partner.onmschina.cn（azure 默认的）。

## 3. 测试结果

如果本地用户后缀为 @azuretest.com azure 端有一个与本地相同用户名称的用户的话。本地同步到 Azure 的用户名称会自动添加一个数字以便作为区分

如果本地用户后缀为 @azuretestbat.com azure 端有一个与本地相同用户名称的用户的话。执行同步后，会同步成本地用户信息。也就是说您在 AD 用户界面看到的“源自”信息，会从 Microsoft Azure Active Directory 改为本地 Active Directory。

进一步排查用户 Azure AD 的配置发现，用户在之前的配置中，禁用了 EnableSoftMatchOnUpn，请参考图 13.2-12。

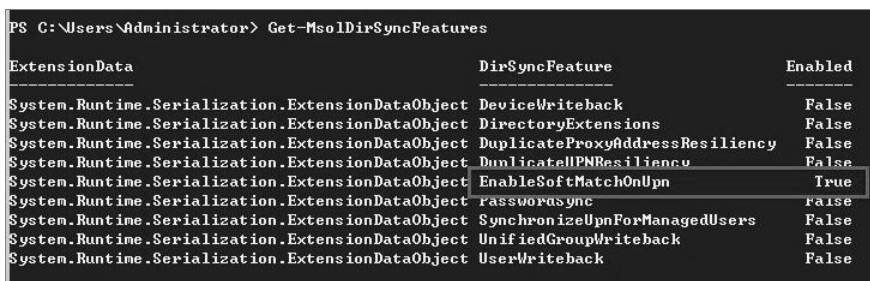


图 13.2-12

解决方案：启用用户 Azure AD 软匹配功能，用户问题解决：

(1) 使用 powershell AD 模块登录中国区 azure：

```
Connect-MsolService -AzureEnvironment azurechinacloud
```

(2) 用软匹配：

```
Set-MsolDirSyncFeature -Feature EnableSoftMatchOnUpn -Enable $true
```

13.2.5 本地与 Azure 做目录同步，属性同步有误

用户想实现本地用户指定的属性同步到 Azure AD 中。

(1) 首先配置好本地的域环境，建议在一台非域控的成员服务器上安装 Azure AD Connect。

(2) 配置好 Azure AD Connect 后，启动该程序，并选择“自定义同步选项”，请参考图 13.2-13。



图 13.2-13

(3) 下一步，在可选功能中，选择“Azure AD 应用和属性筛选”，请参考图 13.2-14。



图 13.2-14

(4) 在 Azure AD 属性中, 勾选“我要进一步限制导出到 Azure AD 的属性”后, 进一步选择同步到 Azure AD 的属性即可, 请参考图 13.2-15。



图 13.2-15

(5) 测试一个本地用户, 该用户的属性如下, 请参考图 13.2-16。



图 13.2-16



（6）执行本地到 Azure AD 的目录同步，查看 Azure 中用户属性（未将“国家或地区”“省/自治区/直辖市”“街道地址”等信息从本地同步到 Azure AD），请参考图 13.2-17。

←

test09

办公电话

选择国家或区域

分机

手机号码

选择国家或区域

街道地址

城市

test

省/自治区/直辖市

邮政编码

11111111

国家或地区

身份验证联系信息

图 13.2-17

## 第十四章 资源组与资源管理器

为了便于资源的管理与组合，Azure 推出了新的资源管理器模式，在这个模式下很多服务的配置和使用方法有所不同，因此，本章针对资源管理器模式进行了详细介绍，将经典模式与资源管理器模式进行了对比说明，并结合实际案例，针对资源管理器模式下的专线服务，负载均衡配置，可用性集，应用程序网关，备份服务以及虚拟机规模集（VMSS，Virtual Machine Scale Set）的配置方法进行了详细的说明。

### 14.1 资源管理器模式

#### 14.1.1 资源管理器模式简介

通常情况下，用户部署在 Azure 中的应用由很多不同类型的组件组成，例如云服务、虚拟机、存储账号、虚拟网络、WebApp、数据库等等，这些组件彼此之间相互引用，相互关联，最终形成一个完整的应用环境。

资源管理器模式中引入了资源组的概念，通过资源组，用户可以将不同类型的组件，或者说资源放入相同的资源组中，以便统一进行维护、管理、升级和监控，甚至是清理和删除。

在资源管理器模式下，所有的部署，请求都不再基于 xml，而是使用了更轻量级的 json 进行数据传递，基于 json 模板，用户可以更高效地部署环境，通过简单地修改模板中的结构和参数，可以更灵活地快速构建大量复杂的部署和应用。

资源管理器模式下，提供了新的安全、审计、标签功能，以方便和灵活地满足用户对于权限控制，资源管理的不同需要。

以下是资源管理器模式中涉及到的几个重要的概念。

**资源（Resource）**：在资源管理器模式中，一个独立可管理的对象就称为一个资源，例如一台虚拟机、一个虚拟网络、一个存储账号、一块虚拟网络接口、一个公共 IP 地址等等。

**资源组（Resource Group）**：资源组是资源的一个容器，同一个资源组内的资源通常是彼此关联或者是有相互引用关系的，通常也只有同一个资源组内的资源彼此之间可以进行组合搭配，形成更为复杂的部署环境。

**资源提供模块（Resource Provider）**：资源管理器模式下，不同的资源的创建和管理请求由不同的资源提供模块来处理，常见的资源提供模块如 Microsoft.Compute，负责处理计算资源相关的请求，Microsoft.Storage 则负责处理存储方面的相关请求。

**资源管理模板（Resource Manager template）**：一个 json 文件，定义了一个或多个资源的部署方式，以及资源之间彼此的依赖关系。通过模板，可以反复多次使用相同模板来实

现批量部署或者环境重建等工作，同时，结合 Visual Studio 的一些组件，可以很方便地实现“拖拽式”地创建和修改模板。

### 14.1.2 资源管理器模式与经典模式的比较

经典模式经历了很长一段时间，在经典模式下，资源之间彼此独立，没有一个容器可以将各种资源组合在一起，为了便于项目维护，用户不得不人工将各种资源通过各种其他方式标记管理（例如在 Excel 中将同一项目相关的虚拟机，虚拟网络，存储记录在一起以说明这些资源具有联系，以便后期的项目维护管理或者工作交接），在环境搭建的过程中，也不得不耗费大量时间和精力按照正确的顺序进行每个资源的逐个部署，或者使用复杂的脚本进行环境搭建。删除环境的时候，也需要逐个资源进行清理。

在权限控制方面，经典模式中无法针对不同类型的资源进行有效控制，一个协同管理员对任何一个订阅具有权限，即是对这个订阅下所有资源都具有读写权限。

引入资源管理器模式后，很好地解决了上面的问题。通过将相关部署放在同一个资源组中，可以有效地进行项目划分，统一管理，维护，甚至统一删除。通过模板可以更高效地进行部署，通过基于角色的访问控制（RBAC，Role-Based Access Control）可以有效地对资源权限配置和管理。

在部署环境时，需要提前规划确认要使用哪种模式进行部署，经典模式和资源管理器模式在功能和资源组织结构方面有一些比较大的差异，可以根据实际部署的架构选择最为合适的模式，以充分利用 Azure 提供的功能。资源管理器模式可以结合 Azure 新版本管理界面（<https://portal.azure.cn/>）实现图形化的操作，而经典模式中只有部分资源可以通过新版本管理界面进行管理。

关于经典模式和资源管理器模式的一些细节对比，可以参考下面的表 14.1-1。

表 14.1-1

资 源	经典模式	资源管理器模式
云服务	云服务作为虚拟机的容器，为虚拟机提供可用性（集）和负载均衡服务	资源管理器模式下已经没有云服务的概念
虚拟网络	虚拟网络作为创建虚拟机的一个可选项而不是必选项，虚拟网络只能通过 Azure 管理界面部署	虚拟网络是创建虚拟机的必选项，虚拟网络可以通过资源管理器提供的 API，模板进行部署
存储账号	虚拟机使用存储账号存放系统和数据磁盘，此外，存储还可以存放其他文件/消息/表单数据等等	与经典模型中存储账号的使用基本一致
可用性集	对于同一可用性集的虚拟机，平台会通过更新域和故障域对其进行划分，保证更新和故障时同一可用性集中虚拟机的可用性，已存在的虚拟机可以随时加入/修改/退出可用性集	可用性集的用途没有改变，但是虚拟机只能在创建时选择可用性集，已存在的虚拟机不能加入/修改/退出可用性集，在配置负载均衡的时候如果需要指定多台虚拟机，则这些虚拟机需要在相同的可用性集下
地缘组	在区域虚拟网络的概念出现后，地缘组已经弃用	资源管理器模式下已经没有地缘组的概念
负载均衡	云服务可以为虚拟机提供外部负载均衡（4 层），内部负载均衡可以通过创建一个内部负载均衡器来实现	内部和外部负载均衡都通过负载均衡器来实现，内部/外部负载均衡器具有独立的公网/内网 IP 地址

(续表)

资 源	经典模式	资源管理器模式
虚拟 IP 地址(VIP)	云服务中正在运行的实例数大于 1 时, 会获得一个公网 IP 地址 (VIP), 这个 IP 地址默认与外部负载均衡器关联, 可以将这个虚拟 IP 地址保留使用	公网 IP 地址可以设置为静态或动态, 公网 IP 地址可以绑定到负载均衡器, 虚拟机, 虚拟网络网关, 应用程序网关等资源上
保留 IP 地址	可以将云服务的公网 IP (VIP) 保留使用	如果想固定一个公网 IP 地址, 可以将这个公网 IP 地址配置为静态地址
虚拟机公网 IP 地址	可以单独为虚拟机配置一个实例级公网 IP 地址, 这个地址不能固定	资源管理器模式下没有云服务的概念, 所以公网 IP 地址默认就与虚拟机网卡绑定
虚拟机终结点	终结点定义了云服务的公网 IP 地址 NAT 转换为虚拟机内网地址的端口转换规则	虚拟机配置中没有终结点的概念了, 即默认全部终结点都会开放, 如果要指定 NAT 规则, 可以在负载均衡器的 NAT 规则中进行类似配置
DNS 名称	云服务默认会获得一个公网唯一的 DNS 名称, 这个 DNS 名称会解析得到云服务的公网 IP 地址 (VIP)	对于公网 IP 地址, 可以为其配置一个对应的 DNS 名称, 这个 DNS 名称是公网唯一的
网络接口	网络接口是定义在虚拟机网络配置中的	网络接口作为一个独立的资源, 可以与虚拟机进行关联, 网络接口可以与公网 IP 地址, 子网, 网络安全组进行关联

## 14.2 从经典模式迁移到资源管理器模式

### 14.2.1 将保留 IP 地址从经典模式迁移到资源管理器模式

将经典模式下的资源向资源管理器模式下迁移, 需要借助资源提供模块 (Resource Provider) Microsoft.ClassicInfrastructureMigrate 提供的 Azure Powershell 命令, 所以在迁移前, 需要确认 Microsoft.ClassicInfrastructureMigrate 是否已经在当前订阅下注册。另外, 建议安装最新版本的 Azure Powershell 以确保相关命令可用。

打开 Azure Powershell, 使用 Azure 账号分别登录经典模式和资源管理器模式:

```
PS C:\Users\XXX> Login-AzureRmAccount -EnvironmentName AzureChinaCloud
PS C:\Users\XXX> Add-AzureAccount -Environment AzureChinaCloud
```

选择经典模式和资源管理器模式的默认订阅 (即原订阅与目标订阅):

```
PS C:\Users\XXX> Select-AzureRmSubscription -SubscriptionName DemoSub1
PS C:\Users\XXX> Select-AzureSubscription -SubscriptionName DemoSub2
```

完成后, 查看 Microsoft.ClassicInfrastructureMigrate 的注册状态:

```
PS C:\Users\XXX> Get-AzureRmResourceProvider | select ProviderNamespace,
RegistrationState | findstr Microsoft.Clas
sicInfrastructureMigrate
Microsoft.ClassicInfrastructureMigrate Registered
```

上面结果显示这个资源提供模块已经注册。如果显示的是未注册的状态, 或者虽然显

示已经注册，但是后续进行迁移的时候仍然报错“Subscription is not registered for migration.”，则需要使用下面的命令进行注册：

```
PS C:\Users\XXX> Register-AzureRmResourceProvider -ProviderNamespace
Microsoft.ClassicInfrastructureMigrate
```

完成上面的步骤后，在经典模式下创建一个新的保留 IP 用于迁移使用：

```
PS C:\Users\XXX> New-AzureReservedIP -ReservedIPName DemoRip -Location
PS C:\Users\XXX> Get-AzureReservedIP -ReservedIPName DemoRip
ReservedIPName      : DemoRip
Address              : 42.159.229.232
Id                  : 57218f79-7119-4e8a-831d-664f6a3fe97b
Label                :
Location             : China East
State                : Created
InUse                : False
ServiceName          :
DeploymentName        :
VirtualIPName        :
.....
```

完成创建后，看到当前这个保留 IP 没有与任何云服务进行绑定，这种情况下满足迁移的条件，使用下面的命令进行迁移：

```
PS C:\Users\XXX> Move-AzureReservedIP -ReservedIPName DemoRip -Prepare
PS C:\Users\XXX> Move-AzureReservedIP -ReservedIPName DemoRip -Commit
```

上面的命令并没有进行 **Validate**，原因是该保留 IP 地址是新创建的，不存在依赖关系，所以直接进行了迁移。对于订阅下已存在的保留 IP 地址，原则上需要首先使用下面的命令进行迁移前的验证工作：

```
PS C:\Users\XXX> (Move-AzureReservedIP -ReservedIPName DemoRip -Validate)
.Result
```

如果该保留 IP 地址符合迁移条件，则会输出下面的结果：

```
Validation Passed. Please see ValidationMessages object for a list of
resources that will be migrated.
```

如果不符合迁移条件，则输出：


```
Validation Failed. Please see ValidationMessages object for additional
details.
```

要进一步查看 **Validate** 失败原因，可以查看命令返回结果中的 **ValidationMessages** 信息：

```
PS C:\Users\XXX> (Move-AzureReservedIP -ReservedIPName DemoRip -Validate)
.ValidationMessages.Message
Reserved IP: DemoRip is assigned to an existing deployment. HostedServiceName:
DemoCloudService DeploymentName: DemoVM. Reserved IP's associated with
```

deployments are migrated as part of deployment/vnet migration. Please use deployment/vnet migration operations to migrate the deployment along with the reserved Ip.

迁移完成后，经典模式下的原保留 IP 地址会自动被清除，在资源管理器模式下会生成一个 IP 地址与原地址相同的静态公网 IP 地址，如图 14.2-1 所示。



公共 IP 地址				
Microsoft				
+ 添加    ≡ 列    ↻ 刷新				
订阅: 选择了 1 个 (共 6 个) - 看不到订阅? 切换目录				
1 个项				
名称	资源组	位置	分配	IP 地址
DemoRip	DemoRip-Migrated	中国东部	静态	42.159.229.232

图 14.2-1

这个 IP 地址会被放到一个新创建的资源组中，资源组名称会在原 IP 名称后添加 -Migrated。

## 14.2.2 将虚拟机从经典模式迁移到资源管理器模式

虚拟机的迁移要根据不同的情况进行区分，经典模式下，虚拟机可以在虚拟网络中创建，也可以不在虚拟网络下。

对于在虚拟网络中的虚拟机的迁移，可以通过迁移虚拟网络来将虚拟机同时迁移到资源管理器模式下，目前不支持单独迁移此类虚拟机。

对于未在虚拟网络中的虚拟机，可以通过下面的方法进行迁移，迁移前建议对虚拟机进行关机操作如下。

首先验证迁移是否可行：

```
PS C:\Users\XXX> $result = Move-AzureService -ServiceName "DemoCloudService"
-DeploymentName "DemoDeployment" -CreateNewVirtualNetwork -Validate
```

如果通过验证，则可以进一步进行 Prepare 操作，如果希望在目的端创建新的虚拟网络，则使用命令。

```
PS C:\Users\XXX> Move-AzureService -ServiceName "DemoCloudService"
-DeploymentName "DemoDeployment" -CreateNewVirtualNetwork -Prepare
```

如果希望使用已有的虚拟网络，使用命令：

```
PS C:\Users\XXX> Move-AzureService -ServiceName "DemoCloudService"
-DeploymentName "DemoDeployment" -UseExistingVirtualNetwork -VirtualNetwork
ResourceGroupName "DemoResourceGroup" -VirtualNetworkName "DemoVNET"
-SubnetName "Subnet-1" -Prepare
```

Prepare 操作成功后，使用下面的命令进行虚拟机迁移：

```
PS C:\Users\XXX> Move-AzureService -ServiceName " DemoCloudService "
-DeploymentName " DemoDeployment " -CreateNewVirtualNetwork -Commit
```

如果需要重新调整 Prepare 的配置，可以使用下面的命令取消 Prepare 的配置：

```
PS C:\Users\XXX> Move-AzureService -ServiceName " DemoCloudService "
-DeploymentName " DemoDeployment " -Abort
```

### 14.2.3 将虚拟网络从经典模式迁移到资源管理器模式

在迁移虚拟网络前，需要保证虚拟网络中没有任何虚拟机的接口被占用。首先进行虚拟网络迁移的 Validate：

```
PS C:\Users\XXX> Move-AzureVirtualNetwork -VirtualNetworkName DemoVNET
-Validate
```

Validate 成功后，执行 Prepare 操作：

```
PS C:\Users\XXX> Move-AzureVirtualNetwork -VirtualNetworkName DemoVNET -
Prepare
```

Prepare 成功后，进行虚拟网络迁移：

```
PS C:\Users\XXX> Move-AzureVirtualNetwork -VirtualNetworkName DemoVNET -
Commit
```

对于配置了 VPN 或者 ER 的虚拟网络，目前不支持使用命令进行迁移。

### 14.2.4 将存储账号从经典模式迁移到资源管理器模式

在迁移存储账号前，需要保证存储账号下的 VHD 文件没有被引用（例如没有关联到现有磁盘或者虚拟机）。使用下面的命令进行 Validate：

```
PS C:\Users\XXX> Move-AzureStorageAccount -StorageAccountName demostorage
-Validate
```

Validate 成功后，进行 Prepare 操作：

```
PS C:\Users\XXX> Move-AzureStorageAccount -StorageAccountName demostorage
-Prepare
```

最后进行 Commit 操作：

```
PS C:\Users\XXX> Move-AzureStorageAccount -StorageAccountName demostorage
-Commit
```

### 14.2.5 关于资源迁移的补充说明

目前通过 Azure Powershell 迁移资源的限制比较多，基本都要求所迁移的资源与其他

资源不互相依赖或者不存在引用关系，所以对于在经典模式下部署复杂的环境或者比较紧凑的环境部署，上面的迁移方法并不适用。

目前的迁移命令都在新版本的 Azure Powershell 中才提供，很多命令执行时还会遇到一些内部报错无法解决，例如在进行 ReservedIP 迁移的时候，Commit 方法遇到报错“The server encountered an internal error”，结果虽然 IP 迁移到资源管理器模式下了，但是经典模式下的原 ReservedIP 对象并没有自动被清理掉，也无法通过手动删除。在迁移虚拟网络和虚拟机的操作中，遇到很多内部错误无法解决，尝试了多个版本的 Azure Powershell，仅有部分操作成功完成，迁移成功。

综上，针对生产环境，由于目前命令以及对应功能还不够稳定，所以目前不建议使用命令进行资源迁移，如果从测试角度出发，可以进行一些简单的测试实验。

如果希望将生产环境部署到资源管理器模式下，强烈建议重新参照资源管理器的新架构进行环境的重新规划和部署，对于经典模式中的数据等资源，可以通过 Azure 存储或者其他媒介进行转移。

## 14.3 资源管理器模式的各类资源

### 14.3.1 虚拟机的使用

在资源管理器模式下，就虚拟机系统本身而言，与经典模式中的虚拟机基本一致，区别在于虚拟机作为资源与其他资源关联的方式有所改变，具体可以参考前面小节中关于经典模式和资源管理器模式的区别介绍。

在资源管理器模式下创建一台新的虚拟机，可以参考下面的基本创建步骤：

首先登录到新版本管理门户中，在左侧服务别表中，选择**虚拟机**，如图 14.3-1 所示。

在展开的虚拟机管理界面中，单击上方的**添加**按钮，如图 14.3-2 所示。



图 14.3-1



图 14.3-2

在弹出的映像选择列表中，选择需要的映像（如果推荐的项目中没有需要使用的映像，可以单击右上角的**更多**，以查看全部可用的映像，如图 14.3-3 所示。



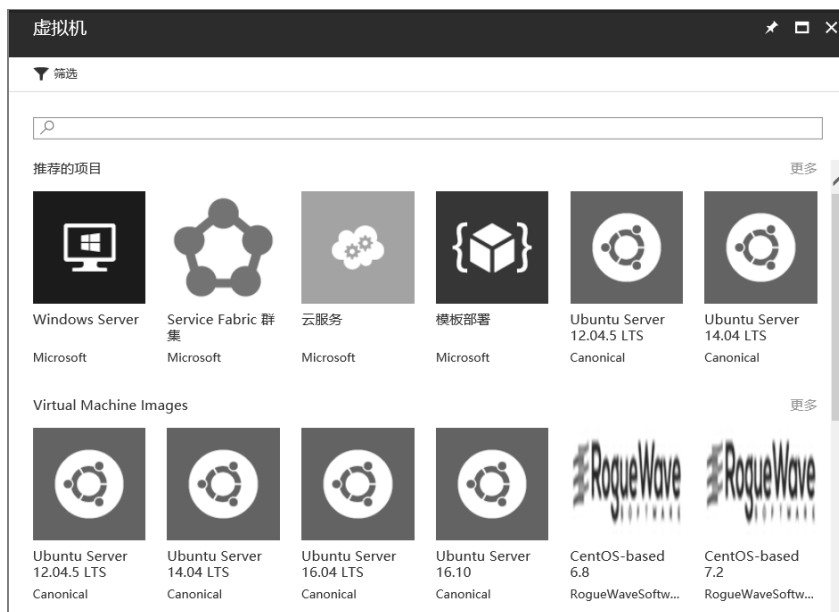


图 14.3-3

选择需要使用的映像后，在展开的菜单中选择要使用的模型，要创建资源管理器模式的虚拟机，所以部署模型选择 **Resource Manager**，如图 14.3-4 所示。



图 14.3-4

单击创建后，在弹出的菜单中填写虚拟机的基本配置，如图 14.3-5 所示。

1 基本  
配置基本设置

2 大小  
选择虚拟机大小

3 设置  
配置可选功能

4 摘要  
Windows Server 2012 R2 Da...

\* 名称

DemoVM

VM 磁盘类型

HDD

\* 用户名

demouser

\* 密码

.....

\* 确认密码

.....

订阅

\* 资源组

☒ 新建 ☐ 使用现有项

DemoResourceGroup

确定

图 14.3-5

磁盘类型可以选择 HDD 或者 SSD。资源组可以选择新建一个资源组，将虚拟机放置到这个新创建的资源组中，也可以选择已经存在的资源组。

确认参数无误后，单击确定，在下一个菜单中选择虚拟机大小，如图 14.3-6 所示。

1 基本  
完成

2 大小  
选择虚拟机大小

3 设置  
配置可选功能

4 摘要  
Windows Server 2012 R2 Da...

选择大小  
浏览可用大小及其功能

显示的价格是以你的当地货币为单位的估价，仅反映 Azure 基础结构费用及适用于该订阅和位置的任何折扣。此价格不包含任何适用的软件成本。推荐的大小根据硬件和软件要求由选定映像的发布服务器决定。

★ 推荐的项目 | 查看所有

D1_V2 标准	D1 标准	A1 标准
1 核心	1 核心	1 核心
3.5 GB	3.5 GB	1.75 GB
2 数据磁盘	2 数据磁盘	2 数据磁盘
2x500 最大 IOPS	2x500 最大 IOPS	2x500 最大 IOPS
50 GB 本地 SSD	50 GB 本地 SSD	负载均衡
负载均衡	负载均衡	
394.32 CNY/月(估计)	357.12 CNY/月(估计)	267.84 CNY/月(估计)

图 14.3-6

• 322 •

如果推荐的项目中没有需要使用的虚拟机型号，可以单击右上角的**查看所有**在所有型号中进一步选择。

在下一步中进一步选择存储账号等设置，如图 14.3-7 所示。

创建虚拟机

1 基本 完成 ✓

2 大小 完成 ✓

3 设置 配置可选功能 >

4 摘要 Windows Server 2012 R2 Da... >

设置

存储

\* 存储帐户 ①  
(新) demoresourcegroup161 >

网络

\* 虚拟网络 ①  
(新) DemoResourceGroup-vnet >

\* 子网 ①  
default (10.2.8.0/24) >

\* 公共 IP 地址 ①  
(新) DemoVM-ip >

\* 网络安全组(防火墙) ①  
(新) DemoVM-nsg >

扩展

扩展 ①  
无扩展 >

高可用性

\* 可用性集 ①  
无 >

正在监视

启动诊断 ①  
已禁用 已启用

来宾操作系统诊断 ①  
已禁用 已启用

\* 诊断存储帐户 ①  
(新) demoresourcegroupdiag859 >

确定

图 14.3-7

存储账号可以选择已有的存储账号，也可以新建一个存储账号。虚拟网络，公网 IP 地址，网络安全组同样可以新建或选择已有项。如果选择新建网络安全组，则默认会根据虚拟机的操作系统类型添加一条入站放行规则，如果是 Linux 虚拟机，则放行 22 端口，如果是 Windows 虚拟机，则放行 3389 端口。

扩展可以选择需要为虚拟机安装扩展，也可以在虚拟机创建后，通过 VM Agent 为虚拟机安装扩展。

可用性集默认不会添加，虚拟机只能在创建的时候进行可用性集的指定，如果未指定，

不能在虚拟机创建后进行修改，所以需要在创建前做好规划。

监视和诊断选项提供了虚拟机性能数据监视和统计功能，启用诊断需要指定一个额外的存储账号，Azure 会将诊断信息和统计数据保存到该指定的存储账号下。

确认参数无误后，单击确认，在最后一步中查看虚拟机的全部设置摘要，如图 14.3-8 所示。

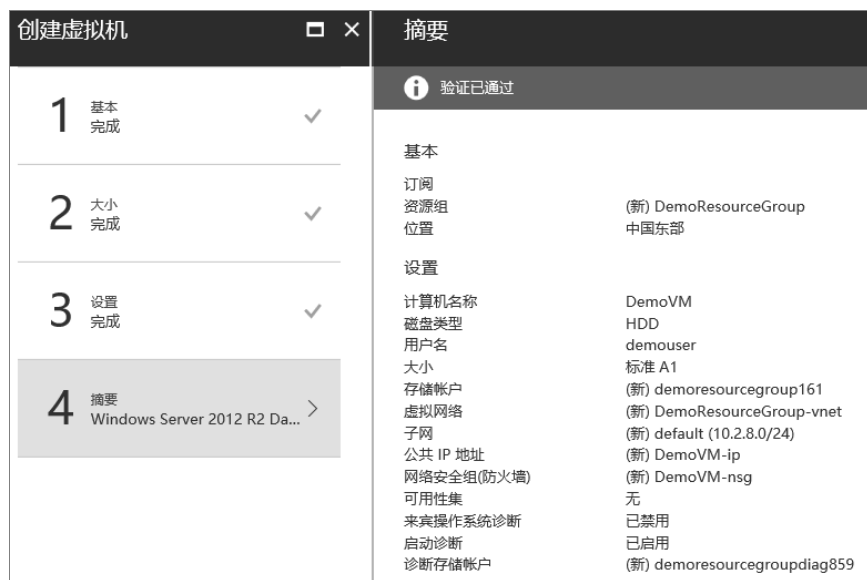


图 14.3-8

确认无误后，单击确定，开始创建虚拟机。

虚拟机创建完成后，就可以在虚拟机列表中看到对应的虚拟机信息了，如图 14.3-9 所示。



图 14.3-9

对于资源管理器模式下的 Windows 虚拟机，默认创建出来只配置了 RDP 服务用于远程访问，经典模式中默认安装的 Powershell 服务并没有启用。

可以通过下面的步骤为资源管理器模式下的 Windows 虚拟配置 Powershell 远程访问：

1) 登录到这台虚拟机中。

- 2) 打开虚拟机的 Powershell 命令行工具, 输入下面的命令允许 Powershell 远程访问:

```
PS C:\Users\XXX> Enable-PSRemoting -Force
WinRM is already set up to receive requests on this computer.
WinRM has been updated for remote management.
Configured LocalAccountTokenFilterPolicy to grant administrative rights
remotely to local users.
```

- 3) 确认 Powershell 端口是否已经侦听 (默认侦听端口是 5985):

```
PS C:\Users\XXX> netstat -ano | findstr 5985
TCP        0.0.0.0:5985          0.0.0.0:0             LISTENING      4
TCP        [::]:5985            [::]:0                 LISTENING      4
```

- 4) 关闭虚拟机的防火墙, 或者添加出入站规则, 放行 5985 端口

5) 如果虚拟机的网络接口配置了网络安全组, 需要在安全组中添加 5985 端口的入站放行规则

6) 在本地计算机 (客户端) 中以管理员权限打开 Powershell, 继续执行下面的语句添加对于虚拟机地址的信任, 其中 42.159.144.73 是要连接的虚拟机的公网 IP 地址:

```
PS C:\Windows\system32> winrm quickconfig
WinRM is not set up to receive requests on this machine.
The following changes must be made:
Start the WinRM service.
Set the WinRM service type to delayed auto start.
Make these changes [y/n]? y
.....
PS C:\Windows\system32> Set-item wsman:localhost\client\trustedhosts
-value 42.159.144.73
WinRM Security Configuration.
This command modifies the TrustedHosts list for the WinRM client. The
computers in the TrustedHosts list might not be authenticated. The client might
send credential information to these computers. Are you sure that you want to
modify this list? [Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
```

7) 配置完成后, 在本地 (客户端) Powershell 中使用下面的命令测试是否能够正常通过 Powershell 端口连接到虚拟机:

```
PS C:\Windows\system32> Enter-PsSession -ComputerName 42.159.144.73 -port
5985 -Authentication Negotiate -Credential demouser -SessionOption
(New-PSSessionOption -SkipCACheck -SkipCNCheck)
```

在弹出的认证窗口中输入密码后, 会发现已经进入虚拟机的命令行了:

```
[42.159.144.73]: PS C:\Users\demouser\Documents>
```

## 14.3.2 存储

资源管理器模式下, 存储的使用和管理方式与经典模式类似, 可以参考下面的步骤在新版本管理门户中创建一个存储账户:

1) 在左侧项目中选择存储账户，在展开的界面上方单击**添加**按钮添加一个新的存储账户，如图 14.3-10 所示。

2) 在弹出的配置界面中，填写创建存储所需的参数，如图 14.3-11 所示。

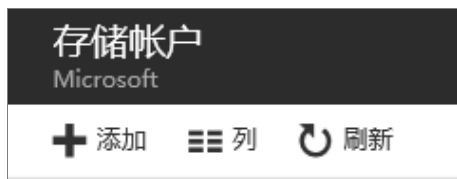


图 14.3-10

图 14.3-11

3) 重要参数说明：

存储账户**名称**必须在 Azure 平台具有唯一性（因为存储账户会对应一个公网的域名，这个域名在公网具有唯一性）。

**部署模型**定义了存储账户是要创建在经典模式中还是资源管理器模式中，这里选择资源管理器模式（Resource Manager）。

**帐户类型**分为“常规用途”和“Blob 存储”，帐户类型在存储账户创建后不能修改。常规用途的存储账号下仍然定义了 4 种类型的结构（Tables, Queues, Files, Blobs），与经典模型下的存储账户相同。对于高级存储账户而言，帐户类型必须选择常规用途。Blob 存储是一种特殊类型的存储账户，用于存放未结构化的数据类型，Blob 存储目前仅支持

“Block Blob”和“Append Blob”（由于高级存储用于存放虚拟机的 VHD，而 VHD 文件是“Page Blob”类型，Blob 存储不支持“Page Blob”，所以高级存储的账户类型必须选择常规用途）。

对于 Blob 存储的存储账户，会多出一个选项“访问层”，如图 14.3-12 所示。

冷：表示 Blob 存储中的数据不会被经常读写，这个选项用于帮助降低数据的“存储成本”

热：表示 Blob 存储中的数据会被经常读写，这个选项用于帮助降低数据的“访问成本”



图 14.3-12

性能用来设置该存储账户是普通存储还是高级存储，这个选项也不能在创建后更改

复制用于配置存储的冗余方式，参考链接：<https://azure.microsoft.com/en-us/documentation/articles/storage-redundancy/>

存储服务加密用于保护存储中的静态数据，当数据写入存储账户后，Azure 平台会对存储的数据进行加密，在访问该数据时，会自动进行解密。参考链接：<https://docs.microsoft.com/en-us/azure/storage/storage-service-encryption>

创建成功后，可以在存储账户列表中查看刚刚创建的存储账户，如图 14.3-13 所示。



图 14.3-13

单击存储账户名称，可以进入存储账户的配置界面，可以看到摘要信息，如图 14.3-14 所示。



图 14.3-14

资源通用信息，如图 14.3-15 所示。

存储设置，如图 14.3-16 所示。

访问密钥：可以查看/重新生成用于管理存储账号的访问密钥。

配置：可以修改存储的配置，例如冗余选项等。

属性：查看存储账号的详细属性。

Blob 服务，如图 14.3-17 所示。



图 14.3-15



图 14.3-16



图 14.3-17

容器：用于查看和管理存储账号下的全部容器以及其中的 Blob 文件。

自定义域：可以为存储账号 Blob 服务指定一个自定义域名作为存储中 Blob 文件访问域名。

文件服务，如图 14.3-18 所示。

表服务，如图 14.3-19 所示。



图 14.3-18



图 14.3-19

队列服务，如图 14.3-20 所示。

监视，如图 14.3-21 所示。



图 14.3-20



图 14.3-21



开启存储的诊断功能，如图 14.3-22 所示。

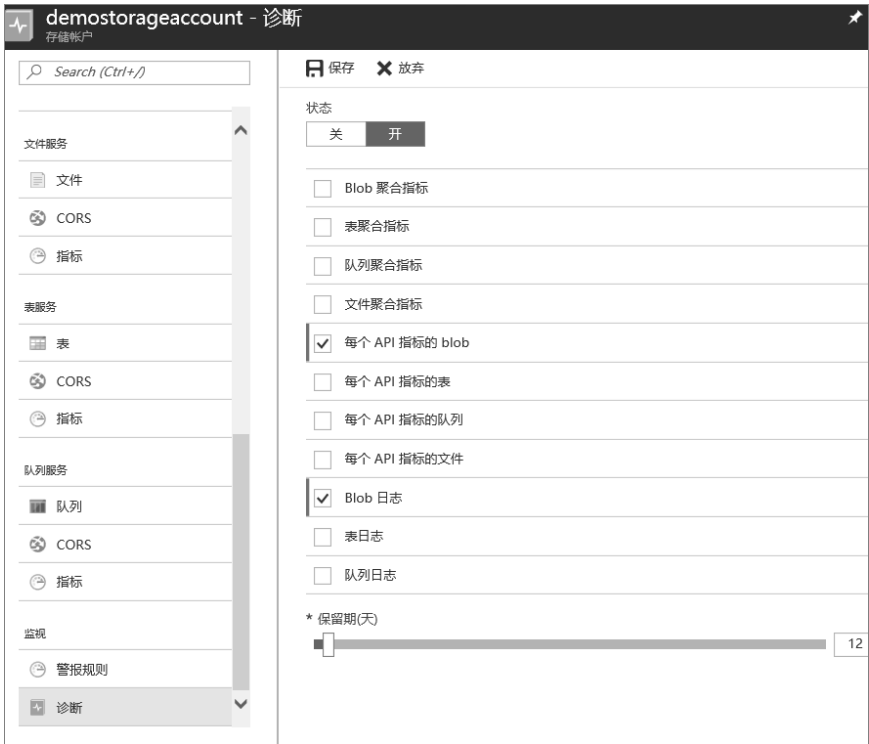


图 14.3-22

启用诊断后，可以在概述中查看选择的诊断数据，还可以编辑图表，如图 14.3-23 所示。

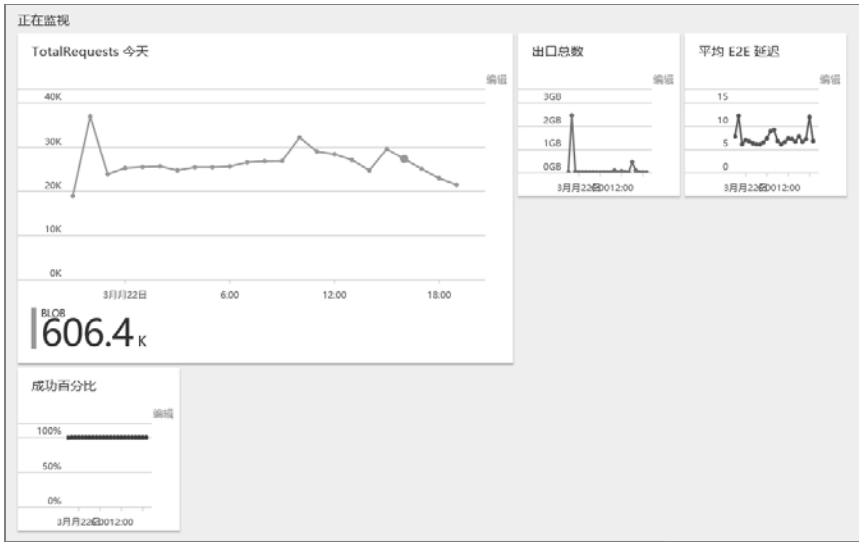


图 14.3-23

还可以针对诊断数据添加警报规则，如图 14.3-24 和图 14.3-25 所示。

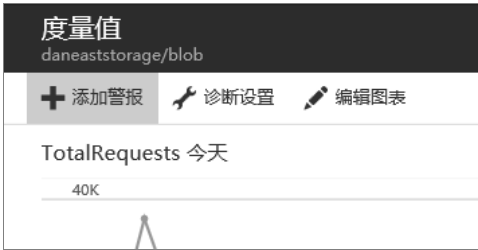


图 14.3-24

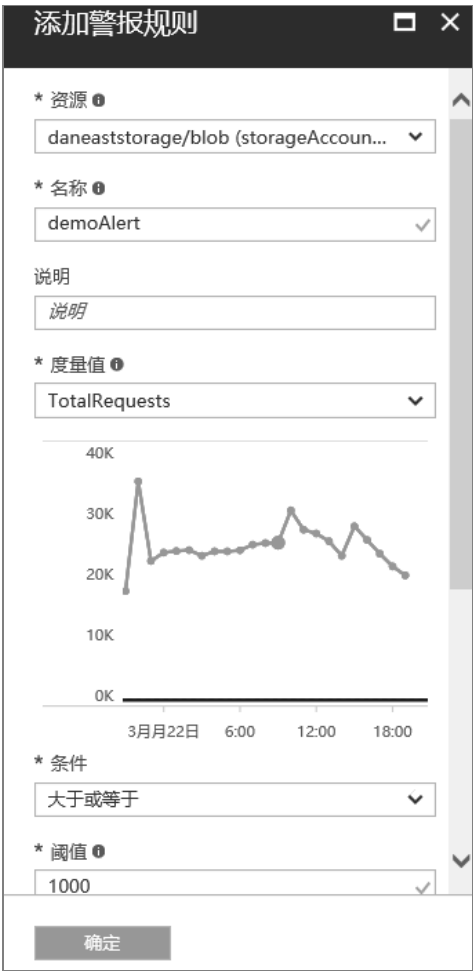


图 14.3-25

关于存储的其他用法，与经典模式下的存储账号用法类似，这里就不再赘述了。

14.3.3 负载均衡

资源管理器模式下，无论内部负载均衡还是外部负载均衡，都是通过负载均衡器来实现的。搭建一个负载均衡器的主要步骤如下（以外部负载均衡为例）：

- 1) 在界面上创建一个负载均衡器，如图 14.3-26 所示。

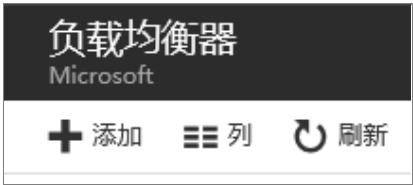


图 14.3-26

填写创建参数，如图 14.3-27 所示。  
需要设置的部分，如图 14.3-28 所示。

创建负载均衡器

\* 名称

demoPublicLB

\* 类型

公共

内部

\* 公共 IP 地址

(新) demoPip

\* 订阅

\* 资源组

新建

使用现有项

DemoResourceGroup

\* 位置

中国东部

设置

前端 IP 池

后端池

运行状况探测

负载均衡规则

入站 NAT 规则

图 14.3-27

图 14.3-28

配置运行状况探测，如图 14.3-29 所示。

添加运行状况探测

demoPublicLB

\* 名称

TCPProbe

协议

HTTP

TCP

\* 端口

80

\* 间隔

5

秒

\* 不正常阈值

2

连续失败

图 14.3-29

添加后端池，注意只有同一个可用性集下的虚拟机能够添加到同一个后端池中，如图 14.3-30 所示。



图 14.3-30

添加**负载均衡规则**，如图 14.3-31 所示。



图 14.3-31

到这里，HTTP 的外部负载均衡器就搭建完成了，后端池中的虚拟机 HTTP 服务启动后，就可以通过负载均衡器的公网 IP 地址进行访问了。

除了提供内部/外部负载均衡的功能外，负载均衡器也可以用于进行 NAT 转换配置（类似于经典模式下云服务的功能），比如对后端池中的虚拟机进行 RDP 端口的映射配置，如图 14.3-32 所示。

添加入站 NAT 规则

demoPublicLB

\* 名称

63389

前端 IP 地址 ⓘ

LoadBalancerFrontEnd (未分配)

IP Version ⓘ

IPv4

服务

RDP

协议

TCP UDP

\* 端口

3389

Associated to

ha (availability set)

Target virtual machine ⓘ

vm2appgatewayforcxwen  
size: Standard\_A1, network interfaces: 1, resource group: RESOURCEGROUPFORBIANAPPGATEWAY

Network IP configuration ⓘ

ipconfig1 (172.22.5.69)

端口映射 ⓘ

默认 自定义

浮动 IP (直接服务器返回) ⓘ

已禁用 已启用

\* 目标端口

3389

图 14.3-32

这样做的好处是，可以不必为虚拟机指定公网 IP 地址，从而避免把虚拟机全部端口暴露在公网中（虽然可以通过网络安全组进行安全加固），提高虚拟机的安全性。同时，将负载均衡器的地址作为统一入口，针对这个入口进行安全配置会更为方便。



### 14.4.2 配置线路

在电信将线路状态置为 Provisioned 状态后，配置 peer 及 VLAN ID。如果线路状态未置为 provisioned 的状态，配置界面将为灰色不可配置，如图 14.4-2 所示。



图 14.4-2

如果如上截图中的线路状态“提供程序状态”为“未设置”时，单击 Azure 公共对等，配置公共对等互联，界面会显示无法配置，如图 14.4-3 所示。



图 14.4-3

14.4.3 创建虚拟网络并为该虚拟网络创建 Express Route 类型的网关

1. 创建 VNET

Azure 门户预览,导航到“虚拟网络”页面，单击“添加”，如图 14.4-4 所示。



图 14.4-4

2. 为虚拟网络创建 Express Route 类型的 VNET 网关

请注意，创建 VNET 网关时，请选择 Express Route 类型。

导航至“虚拟网络网关”页面，单击“添加”，为刚刚创建的 VNET（RMERVNETtest）创建网关，如图 14.4-5 所示。





图 14.4-5

#### 14.4.4 将 VNET 链接到 Express Route 线路

Azure 门户预览导航至连接页面，单击“新建”，如图 14.4-6，在图 14.4-7 中选择对应的虚拟网络网关和 Express Route 线路。



图 14.4-6

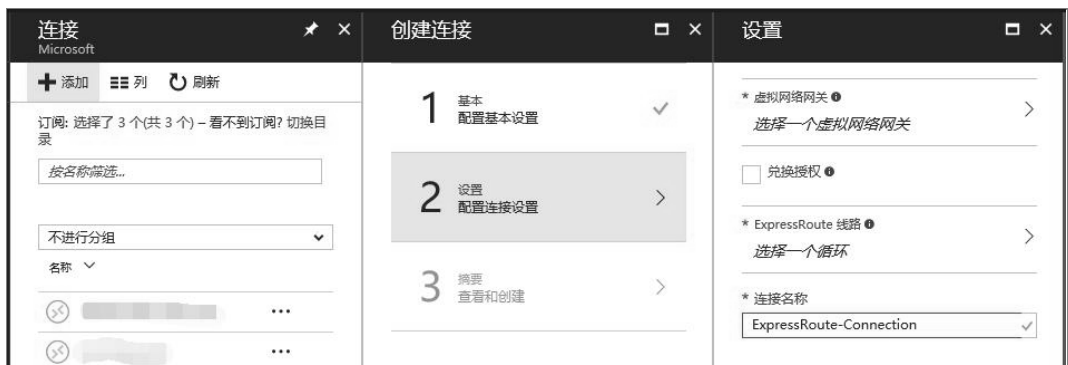


图 14.4-7

## 14.5 如何在 ARM 模式下去部署 ILB 环境

### 1. 该实验需要实现的目标

- 1) 在 Portal 上创建 2 台虚机 (Server1&Server2)，并且这两台虚机是属于同一个可用性集中。
- 2) 创建内部负载均衡器。
- 3) 在内部负载均衡器上为这 2 台虚机 (Server1&Server2) 设置内部负载均衡终结点，如图 14.5-1 所示。

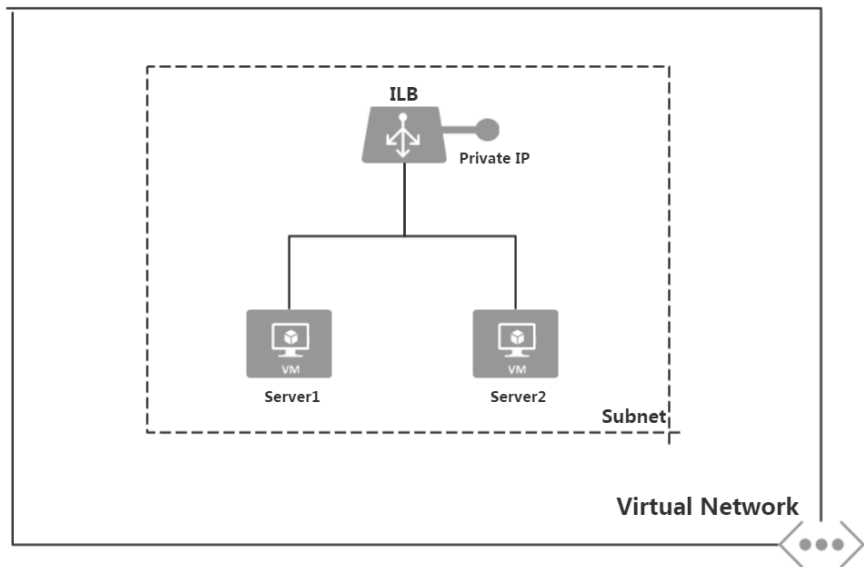


图 14.5-1

### 2. 具体的操作步骤如下

- 1) 在 Portal（门户预览）上的同一个资源组中创建两台虚机，如图 14.5-2 所示。

 Server1	虚拟机
 Server2	虚拟机

图 14.5-2

- 2) 创建一个负载均衡器。
- a) 选择负载均衡器，如图 14.5-3 所示。
- b) 然后单击“添加”，如图 14.5-4 所示。



图 14.5-3

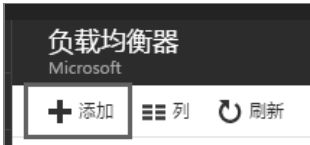


图 14.5-4

- c) 设置负载均衡器参数，然后创建 ILB，如图 14.5-5 所示。
- 为负载均衡器设定一个名称。
- 选择类型：内部。
- 选择虚拟网络要与之前创建的两台虚机处于同一个虚拟网络，同一个子网。
- 建议内部负载均衡器的 IP 地址设置为一个静态 IP 地址，并且该地址不能被占用。
- 选择资源组，并且保证虚机与负载均衡器都部署在同一个资源组中。
- 3) 创建完 ILB 后，选择刚刚创建的负载均衡器，并添加一个“探测”规则。
- a) 在资源组中单击概述，如图 14.5-6 所示。



图 14.5-5



图 14.5-6

b) 选择 ILB 负载均衡器，如图 14.5-7 所示。

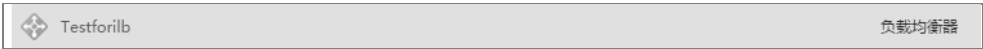


图 14.5-7

- c) 单击“运行状态探测”，如图 14.5-8 所示。
- d) 然后单击“添加”，如图 14.5-9 所示。
- e) 对运行状况探测进行设置，然后创建运行状况探测，如图 14.5-10 所示。
- 设置运行状况探测名称。
  - 设置协议：TCP。
  - 端口：80。
  - 间隔：5 秒（间隔：每次探测间隔的时间）。
  - 不正常阈值：2（不正常阈值：虚机被视为不健康之前探测所尝试的连接总次数）。
  - 全部参数设置好后，单击保存来添加运行状况探测。

- 4) 创建完运行状况探测后，开始运行编辑“后端池”，将两台虚拟机添加到其中。
- a) 添加后端池，如图 14.5-11 所示。



图 14.5-8



图 14.5-9



图 14.5-10



图 14.5-11

b) 把同一个可用性集中的虚机都加入到后端池中，如图 14.5-12 所示。

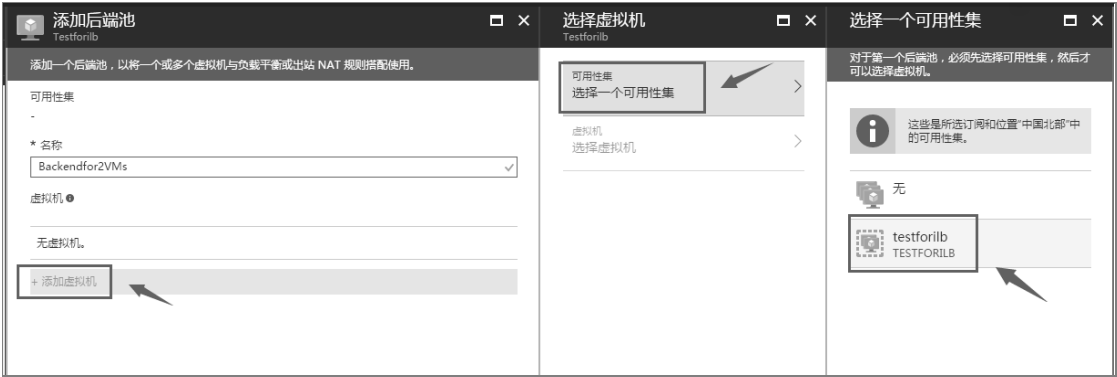


图 14.5-12

- c) 配置完后端池后，添加“平衡均衡规则”。
- d) 单击负载平衡规则，如图 14.5-13 所示。



图 14.5-13

- e) 添加负载平衡规则，如图 14.5-14 所示。
- 选择之前设置好的 ILB 前端 IP 地址。
  - 确认好 ILB 终结点的前端和后端映射端口的关系。
  - 选择之前设置好的后端池 & 运行状况探测。
  - 会话持续性我选择为无（无的意思就是默认情况下使用五元组的负载分发机制。客户端 IP: 二元组。客户端 IP 和协议: 三元组），如图 14.5-15 和图 14.5-16 所示。
  - 空闲超时（分钟）的意思为 TCP 或 HTTP 打开连接中的保活超时时间，该机制不需要客户端来发生保活数据包来维持 TCP 或 HTTP 的连接。

添加负载均衡规则

Testforilb

\* 名称

ILB\_rules

\* 前端 IP 地址

10.2.5.10 (LoadBalancerFrontEnd)

协议

TCPUDP

\* 端口

80

\* 后端端口

80

后端池

Backendfor2VMs

运行状况探测

HTTP80\_Probe (TCP:80)

会话持续性

无

空闲超时(分钟)

4

浮动 IP (直接服务器返回)

已禁用已启用

确定

图 14.5-14

会话持续性

无

无

客户端 IP

客户端 IP 和协议

图 14.5-15

会话持续性指定来自客户端的流量在会话持续期间必须由后端池中的同一虚拟机进行处理。“无”指定来自同一客户端的连续请求可以由任何虚拟机进行处理。“客户端 IP”指定来自同一客户端 IP 地址的连续请求将由同一虚拟机进行处理。“客户端 IP 和协议”指定来自同一客户端 IP 地址和协议组合的连续请求将由同一虚拟机进行处理。

图 14.5-16

## 14.6 两台 ARM 虚拟机的负载均衡配置

可以参考下面的步骤将 ARM 模式下同一个可用性集中的两台虚机加入到外部负载均衡器中。

**注：**创建虚机的时候，请保证两台虚机分别创建在不同的存储账户中。

- (1) 添加可用性集，如图 14.6-1 所示。
- (2) 设定好相关信息，然后创建可用性集，如图 14.6-2 所示。

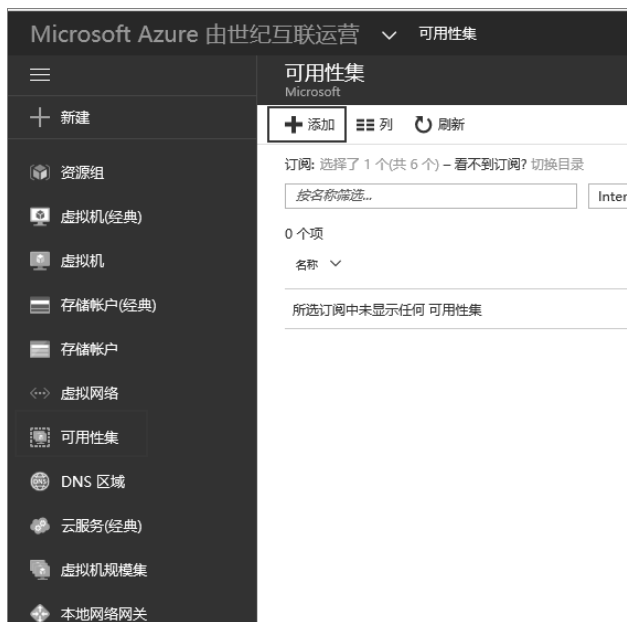


图 14.6-1

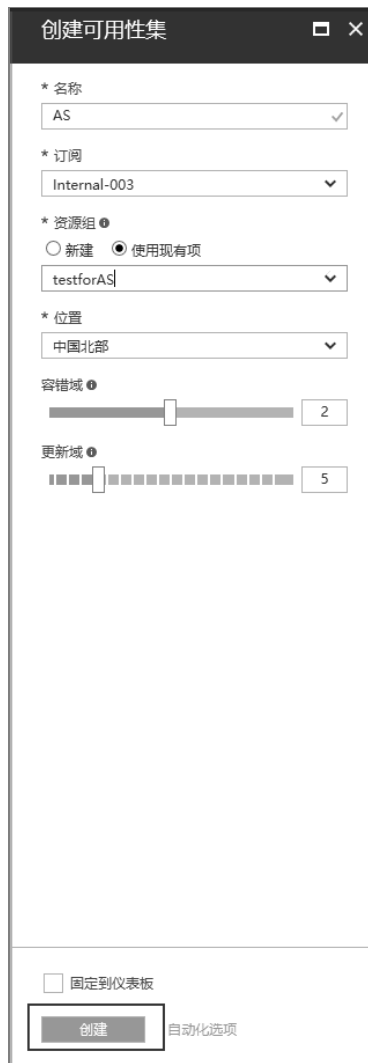


图 14.6-2

(3) 创建第一台虚机，并把其加入到同一个可用性集中。并同时保证第一台虚机与第二台虚机分别创建在不同的存储账户中。

- 填写创建虚机的基本信息，如图 14.6-3 所示。





图 14.6-3

- 配置可选功能: 单击“存储账户”, 新建存储账户名为 rgbymzdisk882, 如图 14.6-4、图 14.6-5 和图 14.6-6 所示。



图 14.6-4



图 14.6-5



图 14.6-6

- 配置可选功能：单击“可用性集”，然后把虚拟机 test1 加入到之前创建好的那个可用性集（AS），如图 14.6-7、图 14.6-8 和 14.6-9 所示。



图 14.6-7



图 14.6-8



图 14.6-9

(4) 创建第二虚机，然后将其加入到同一个可用性集中。并同时保证第一台虚机与第二台虚机分别创建在不同的存储账户中。

填写创建虚机的基本信息，如图 14.6-10 所示。



图 14.6-10

- 设置可选功能: 单击“存储账户”, 新建存储账户名为 rgbymzdisk883。如图 14.6-11、图 14.6-12 和图 14.6-13 所示。

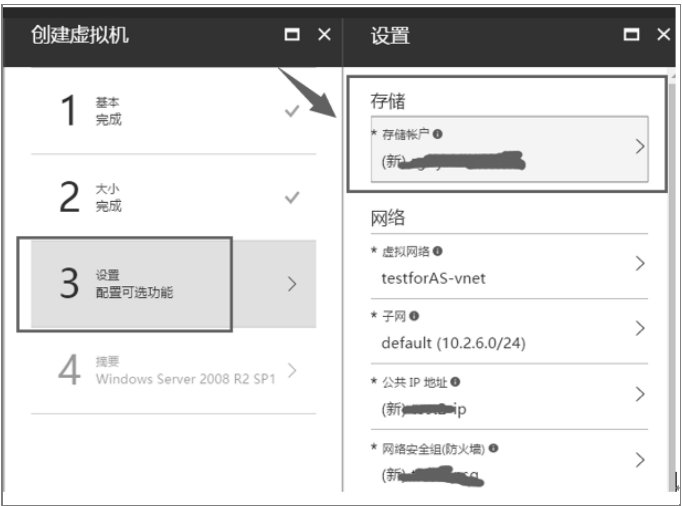


图 14.6-11

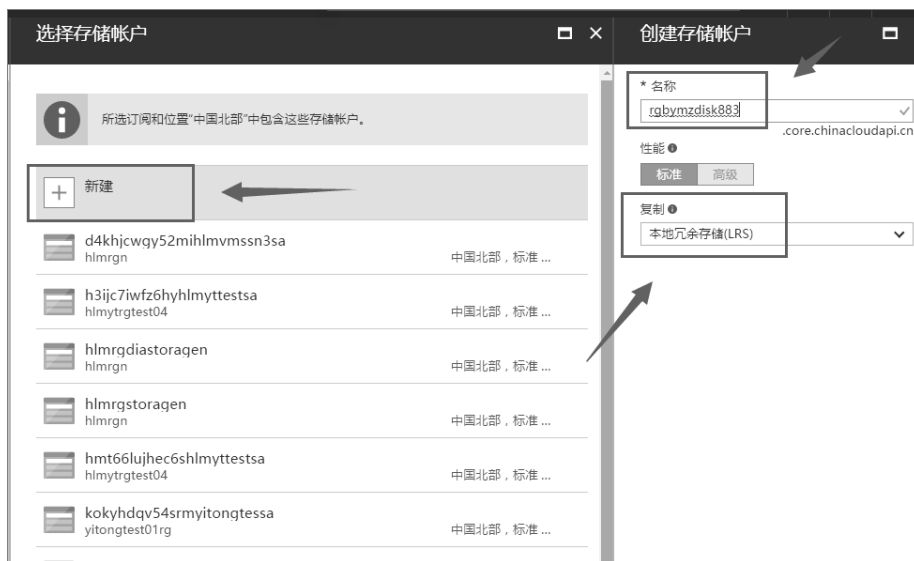


图 14.6-12



图 14.6-13

- 设置可选功能: 单击“可用性集”, 然后把虚拟机 test2 加入到之前创建好的那个可用性集 (AS)。如图 14.6-14、图 14.6-15 和 14.6-16 所示。



图 14.6-14



图 14.6-15



图 14.6-16

(5) 查看可用性集中虚拟机 test1 & test2 所在容错域和更新域名的位置，如图 14.6-17 和图 14.6-18 所示。



图 14.6-17

名称	状态	订阅域	更新域
test1	正在运行	0	0
test2	正在运行	1	1

图 14.6-18

(6) 在资源组中选择负载均衡器（testforAS），如图 14.6-19 所示。



图 14.6-19

(7) 为外部负载均衡器添加公共 IP 地址。

- 单击“前端 IP 池”，如图 14.6-20 所示。



图 14.6-20

- 单击“IP 地址”，如图 14.6-21 所示。



图 14.6-21

- 新建公共 IP 地址，如图 14.6-22 所示。
- 为新建的公共 IP 地址编写一个名称，如图 14.6-23 所示。





图 14.6-22



图 14.6-23

- 单击“保留”来创建新的公共 IP 地址，如图 14.6-24 所示。

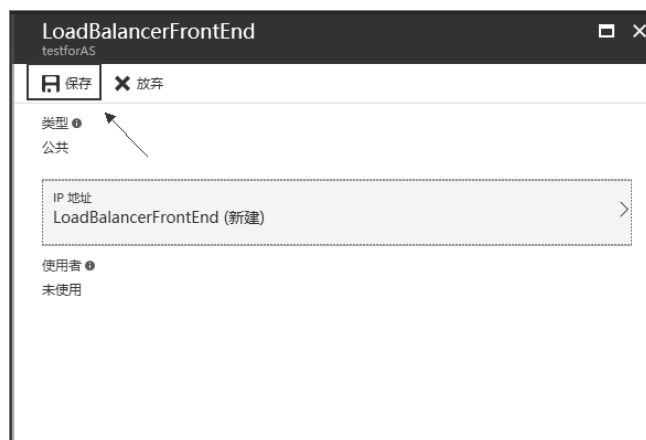


图 14.6-24

- (8) 设置外部负载均衡器的“运行状态探测”，如图 14.6-25 和图 14.6-26 所示。



图 14.6-25



图 14.6-26

(9) 为外部负载均衡器设置后端池，把同一个可用性集（AS）中的两台虚拟机（test1 & test2）加入到负载均衡器的后端池中。

注：只有在同一个可用性集中以及同一个虚拟网络下的虚拟机才能加入到负载均衡集中，并且部署在负载均衡集中的虚拟机是必须创建在同一个可用性集中以及同一个虚拟网络下。

- 单击添加“后端池”，如图 14.6-27 所示。
- 为后端池命名，然后单击“添加虚拟机”，如图 14.6-28 所示。



图 14.6-27

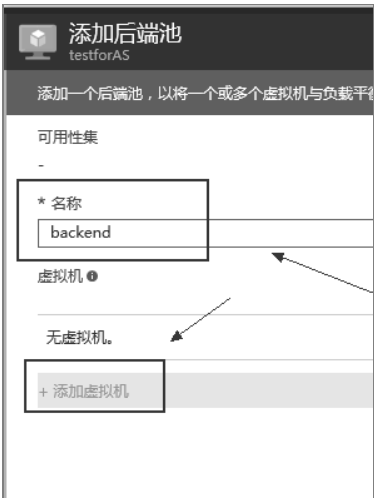


图 14.6-28

- 选择“可用性集”，如图 14.6-29 所示。
- 选择之前创建好的那个可用性集（AS），如图 14.6-30 所示。



图 14.6-29



图 14.6-30

- 然后选择虚拟机 test1 & test2，如图 14.6-31 和图 14.6-32 所示。



图 14.6-31



图 14.6-32

- 保存后端池的配置，如图 14.6-33 所示。



图 14.6-33

(10) 为外部负载均衡器添加“负载平衡规则”，如图 14.6-34 和图 14.6-35 所示。



图 14.6-34



图 14.6-35

## 14.7 应用程序网关

Azure 应用程序网关是服务形式的应用程序传送控制器（ADC），借此为应用程序提供各种第 7 层负载均衡功能。它提供完全由 Azure 管理的高度可用、可缩放的服务。应用程序网关支持 SSL 卸载和端到端 SSL、基于 Cookie 的会话相关性、基于 URL 路径的路由、多站点托管，等等，如图 14.7-1 所示。

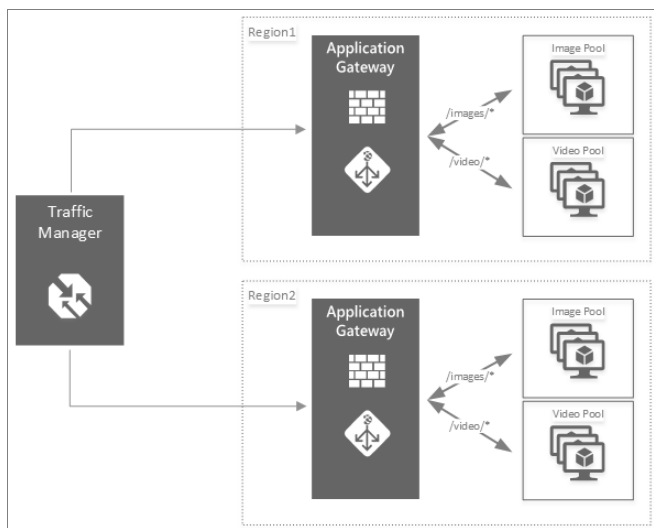


图 14.7-1

应用程序网关当前支持具有以下功能的第 7 层应用程序传送。

- **HTTP 负载均衡**—应用程序网关提供轮循机制负载均衡。负载均衡在第 7 层完成，仅用于 HTTP(S) 流量。快速扩展你的大计算和大数据应用程序。
- **基于 cookie 的会话相关性**—想要在同一后端保留用户会话时，此功能十分有用。借助受网关管理的 cookie，应用程序网关能够将来自用户会话的后续流量转到同一后端进行处理。
- **安全套接字层（SSL）卸载**—此功能让 Web 服务器免于执行解密 HTTPS 流量的高成本任务。通过在应用程序网关终止 SSL 连接，并将请求转发到未加密的服务器，Web 服务器不用承担解密的负担。应用程序网关会重新加密响应，然后再将它发回客户端。
- **端到端 SSL**—应用程序网关支持对流量进行端到端加密。应用程序网关通过在应用程序网关上终止 SSL 连接来完成此任务。网关随后将路由规则应用于流量、重新加密数据包，并根据定义的路由规则将数据包转发到适当的后端。来自 Web 服务器的任何响应都会经历相同的过程返回最终用户。
- **基于 URL 的内容路由**—此功能能够使用不同的后端服务器来处理不同的流量。可将 Web 服务器上的文件夹流量或 CDN 流量路由到不同的后端，让不提供特定内容的后端减少不必要的负载。

- **多站点路由**—应用程序网关允许在单个应用程序网关上合并最多 20 个网站。
- **WebSocket 支持**—应用程序网关的另一个重要功能是对 WebSocket 的本机支持。
- **运行状况监视**—应用程序网关提供默认的后端资源运行状况监视，以及用于监视更多特定方案的自定义探测。

### 14.7.1 配置面向 Internet 的负载均衡

面向 Internet 的负载均衡拓扑图，如图 14.7-2 所示。

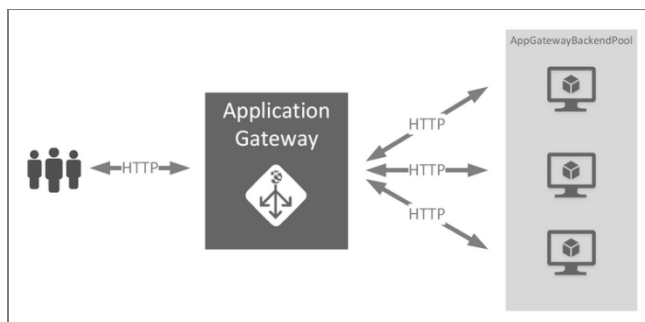


图 14.7-2

(1) 打开 Azure 管理门户，单击“新建”>“网络”>“应用程序网关”，如图 14.7-3 所示。

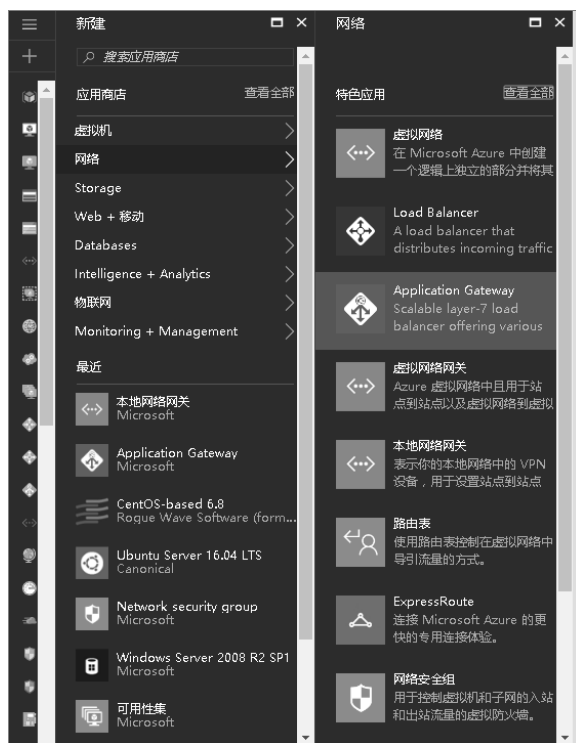


图 14.7-3

(2) 填写应用程序网关相关的基本信息，如图 14.7-4 所示。

图 14.7-4

注意：

进行测试时，可以选择 1 作为实例计数。必须知道的是，2 以下的实例计数不受 SLA 支持，因此不建议使用。小型网关用于开发/测试，不用于生产。

(3) 定义要使用的虚拟网络、面向 Internet 的公共 IP 等，如图 14.7-5 所示。

图 14.7-5

**注意：**

虚拟网络：这里可以选择新建虚拟网络，或者选择现有的虚拟网络，如果选择现有的虚拟网络，请一定预留一个空的子网或只限应用程序网关资源的子网才能使用该虚拟网络。

前端 IP 配置：这里配置应用程序网关的前端 IP，可以选择公共 IP 或者专用 IP。具体取决于应用程序网关是面向 Internet 还是仅供内部使用。本篇选择公共 IP 地址，此时系统会选择自动分配的 IP。

侦听器配置：如果使用的是 http，不需进行任何配置，单击“确定”即可。若要使用 https，需要进一步配置。需要提供证书，由于需要证书的私钥，请提供证书的 .pfx 导出结果以及密码。

(4) 查看“摘要”并单击“确定”开始创建后，进入“应用程序网关”查看详情，如图 14.7-6 所示。



图 14.7-6

(5) 将服务器添加到后端池。

单击“后端池”，然后选择当前的后端池，如图 14.7-7 所示。



图 14.7-7



(6) 在文本框中添加 IP 或者 FQDN 值，然后单击“保存”，如图 14.7-8 所示。



图 14.7-8

此操作会将值保存到后端池。更新应用程序网关后，进入应用程序网关的流量将路由到在此步骤中添加的后端地址。

以上步骤会创建基本的应用程序网关，提供侦听器、后端池、后端 http 设置以及规则的默认设置。预配成功后，即可根据部署修改这些配置。

### 14.7.2 内部负载均衡

面向内部终结点的 Azure 应用程序网关，也称为内部负载均衡器（ILB）。配置使用 ILB 的网关适用于不向 Internet 公开的内部业务线应用程序。对于位于不向 Internet 公开的安全边界内的多层应用程序中的服务和层也很有用，但仍需要执行轮循负载分布、会话粘性或安全套接字层（SSL）终止。

以下是创建应用程序网关所需的步骤（本节使用 Azure PowerShell 方式创建）：

- 创建 Resource Manager 的资源组。
- 为应用程序网关创建虚拟网络和子网。
- 创建应用程序网关配置对象。
- 创建应用程序网关资源。

#### 1. 创建 Resource Manager 的资源组

创建新的资源组（如果要使用现有的资源组，请跳过此步骤）：

```
New-AzureRmResourceGroup -Name appgw-rg -location "China North"
```

Azure Resource Manager 要求所有资源组指定一个位置。此位置将用作该资源组中的

资源的默认位置。请确保用于创建应用程序网关的所有命令都使用相同的资源组。

在上述示例中，我们在位置“中国北部”创建了名为“appgw-rg”的资源组。

## 2. 为应用程序网关创建虚拟网络和子网

以下示例演示如何使用 Resource Manager 创建虚拟网络：

(1) 将地址范围 172.0.0.0/24 分配给用于创建虚拟网络的子网变量：

```
$subnetconfig = New-AzureRmVirtualNetworkSubnetConfig -Name subnet01
-AddressPrefix 172.0.0.0/24
```

(2) 使用前缀 172.0.0.0/16 和子网 172.0.0.0/24，在中国北部区域的“appgw-rg”资源组中创建名为“appgwvnet”的虚拟网络：

```
$vnet = New-AzureRmVirtualNetwork -Name appgwvnet -ResourceGroupName appgw-rg
-Location "China North" -AddressPrefix 172.0.0.0/16 -Subnet $subnetconfig
```

(3) 将子网对象分配到变量\$subnet 以完成后续步骤：

```
$subnet = $vnet.subnets[0]
```

## 3. 创建应用程序网关配置对象

(1) 创建名为“gatewayIP01”的应用程序网关 IP 配置：

```
$gipconfig = New-AzureRmApplicationGatewayIPConfiguration -Name gatewayIP01
-Subnet $subnet
```

当应用程序网关启动时，它会从配置的子网获取 IP 地址，再将网络流量路由到后端 IP 池中的 IP 地址。请记住，每个实例需要一个 IP 地址。

(2) 配置名为“pool01”的后端 IP 地址池，其 IP 地址为“172.0.0.4, 172.0.0.5, 172.0.0.6”：

```
$pool = New-AzureRmApplicationGatewayBackendAddressPool -Name pool01
-BackendIPAddresses 172.0.0.4, 172.0.0.5, 172.0.0.6
```

这些 IP 地址将接收来自前端 IP 终结点的网络流量。替换上述 IP 地址，添加自己的应用程序 IP 地址终结点。

(3) 为后端池中进行了负载均衡的网络流量配置应用程序网关设置“poolsetting01”：

```
$poolSetting = New-AzureRmApplicationGatewayBackendHttpSettings -Name
poolsetting01 -Port 80 -Protocol Http -CookieBasedAffinity Disabled
```

(4) 为 ILB 配置名为“frontendport01”的前端 IP 端口：

```
$fip = New-AzureRmApplicationGatewayFrontendPort -Name frontendport01 -Port
80
```

(5) 创建名为“fipconfig01”的前端 IP 配置，并将其与当前虚拟网络子网中的某个专用 IP 相关联：

```
$fipconfig = New-AzureRmApplicationGatewayFrontendIPConfig -Name fipconfig01
-Subnet $subnet
```

(6) 创建名为“listener01”的侦听器，并将前端端口与前端 IP 配置相关联：

```
$listener = New-AzureRmApplicationGatewayHttpListener -Name listener01
-Protocol Http -FrontendIPConfiguration $fipconfig -FrontendPort $fp
```

(7) 创建名为“rule01”的负载均衡器路由规则，用于配置负载均衡器的行为：

```
$rule = New-AzureRmApplicationGatewayRequestRoutingRule -Name rule01
-RuleType Basic -BackendHttpSettings $poolSetting -HttpListener $listener
-BackendAddressPool $pool
```

(8) 配置应用程序网关的实例大小：

```
$sku = New-AzureRmApplicationGatewaySku -Name Standard_Small -Tier Standard
-Capacity 2
```

注意：

InstanceCount 的默认值为 2，最大值为 10。GatewaySize 的默认值为 Medium。可以在 Standard\_Small、Standard\_Medium 和 Standard\_Large 之间进行选择。

#### 4. 使用 New-AzureApplicationGateway 创建应用程序网关

创建包含上述步骤中所有配置项目的应用程序网关。示例中的应用程序网关名为“appgwtest”。

```
$appgw = New-AzureRmApplicationGateway -Name appgwtest -ResourceGroupName
appgw-rg -Location " China North " -BackendAddressPools $pool
-BackendHttpSettingsCollection $poolSetting -FrontendIpConfigurations
$fipconfig -GatewayIpConfigurations $gipconfig -FrontendPorts $fp
-HttpListeners $listener -RequestRoutingRules $rule -Sku $sku
```

#### 5. 删除应用程序网关

若要删除应用程序网关，请按顺序执行以下步骤：

- 使用 Stop-AzureRmApplicationGatewaycmdlet 停止该网关。
- 使用 Remove-AzureRmApplicationGatewaycmdlet 删除该网关。
- 使用 Get-AzureApplicationGatewaycmdlet 验证是否已删除该网关。

(1) 取应用程序网关对象，并将其关联到变量“\$getgw”：

```
$getgw = Get-AzureRmApplicationGateway -Name appgwtest -ResourceGroupName
appgw-rg
```

(2) 使用 Stop-AzureRmApplicationGateway 停止应用程序网关。

此示例在第一行显示 Stop-AzureRmApplicationGateway cmdlet，接着显示输出：

```
Stop-AzureRmApplicationGateway -ApplicationGateway $getgw
VERBOSE: 9:49:34 PM - Begin Operation: Stop-AzureApplicationGateway
VERBOSE: 10:10:06 PM - Completed Operation: Stop-AzureApplicationGateway
Name      HTTP Status Code  Operation ID      Error
```

```
-----
Successful OK                                ce6c6c95-77b4-2118-9d65-e29defadffb8
-----
```

(3) 应用程序网关进入停止状态后, 请使用 `Remove-AzureRmApplication Gateway cmdlet` 删除该服务。

```
Remove-AzureRmApplicationGateway -Name appgwtest -ResourceGroupName
appgw-rg -Force
VERBOSE: 10:49:34 PM - Begin Operation: Remove-AzureApplicationGateway
VERBOSE: 10:50:36 PM - Completed Operation: Remove-AzureApplicationGateway
Name      HTTP Status Code      Operation ID      Error
-----
Successful OK      055f3a96-8681-2094-a304-8d9a11ad8301
```

(4) 验证是否已删除:

```
Get-AzureRmApplicationGateway -Name appgwtest -ResourceGroupName appgw-rg
VERBOSE: 10:52:46 PM - Begin Operation: Get-AzureApplicationGateway
Get-AzureApplicationGateway : ResourceNotFound: The gateway does not exist.
```

**注意:**

可以使用 `-force` 开关来禁止显示该删除的确认消息。

若要验证是否已删除服务, 可以使用 `Get-AzureRmApplicationGateway cmdlet`。此步骤不是必需的。

### 14.7.3 SSL OffLoad

应用程序网关支持对流量进行端到端加密。应用程序网关通过在应用程序网关上终止 SSL 连接来完成此任务。网关随后将路由规则应用于流量、重新加密数据包, 并根据定义的路由规则将数据包转发到适当的后端。来自 Web 服务器的任何响应都会经历相同的过程返回最终用户。如图 14.7-9 所示的拓扑图。

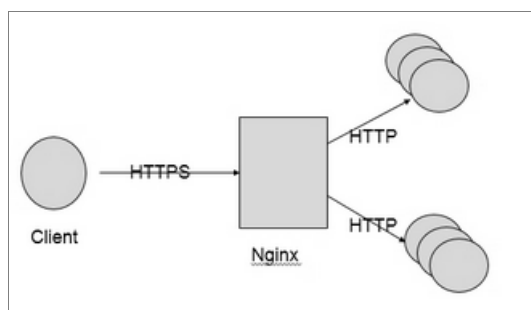


图 14.7-9

**准备工作:**

若要配置应用程序网关的 SSL 卸载, 必须提供证书。此证书在应用程序网关上加载, 用于加密和解密通过 SSL 发送的流量。证书需采用个人信息交换 (pfx) 格式。此文件格式适用于导出私钥, 后者是应用程序网关对流量进行加解密所必需的。

配置 SSL 卸载步骤如下。

1. 添加侦听器

(1) 登录 Azure 门户，然后选择现有的应用程序网关，如图 14.7-10 所示。



图 14.7-10

(2) 单击“侦听器”，然后单击“添加”按钮添加新的侦听器，如图 14.7-11 所示。



图 14.7-11

- **名称**—这是侦听器的友好名称。
- **前端 IP 配置**—这是用于侦听器的前端 IP 配置。
- **前端端口（名称/端口）**—用在应用程序网关前端的端口的友好名称，以及所使用的实际端口。
- **协议**—一个开关，用于确定为前端使用了 https 还是 http。
- **证书（名称/密码）**—如果使用了 SSL 卸载，则需对此设置使用 .pfx 证书，并需提供友好名称和密码。

## 2. 创建规则并将其关联到侦听器

(1) 创建侦听器后，此时需创建规则来处理来自侦听器的流量

单击应用程序网关的“规则”，然后单击“添加”，如图 14.7-12 所示。



图 14.7-12

(2) 在“添加基本规则”变边栏选项卡中，键入规则的名称并选择上一步创建的侦听器。以及适当的后端池和 http 设置，然后单击“确定”，如图 14.7-13 所示。

- **名称**—这是可在门户中访问的规则友好名称。
- **侦听器**—这是用于规则的侦听器。
- **默认后端池**—此设置可定义要用于默认规则的后端
- **默认 HTTP 设置**—此设置可定义要用于默认规则的 HTTP 设置。

保存完以后，应用程序网关将负责处理流量的加解密。应用程序网关和后端 Web 服务器之间的所有流量将通过 http 来处理。任何通过 https 启动的需要返回到客户端的通信都会返回到加密的客户端。

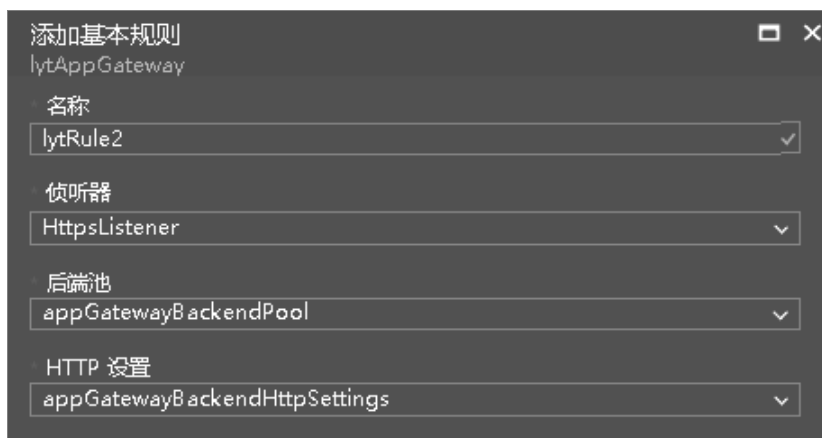


图 14.7-13

#### 14.7.4 基于 URL 的访问设置

基于 URL 路径的路由，可根据 Http 请求的 URL 路径来关联路由。它将检查是否有路由连接到针对应用程序网关中的 URL 列表配置的后端池，并将网络流量发送到定义的后端池。基于 URL 的路由的常见用法是将不同内容类型的请求负载均衡到不同的后端服务器池。

基于 URL 的路由将新的规则类型引入应用程序网关。应用程序网关有两种规则类型：基本规则和基于路径的规则。基本规则类型针对后端池提供轮循机制服务，而基于路径的规则除了轮循机制分发以外，还在选择后端池时考虑请求 URL 的路径模式，如图 14.7-14 所示。

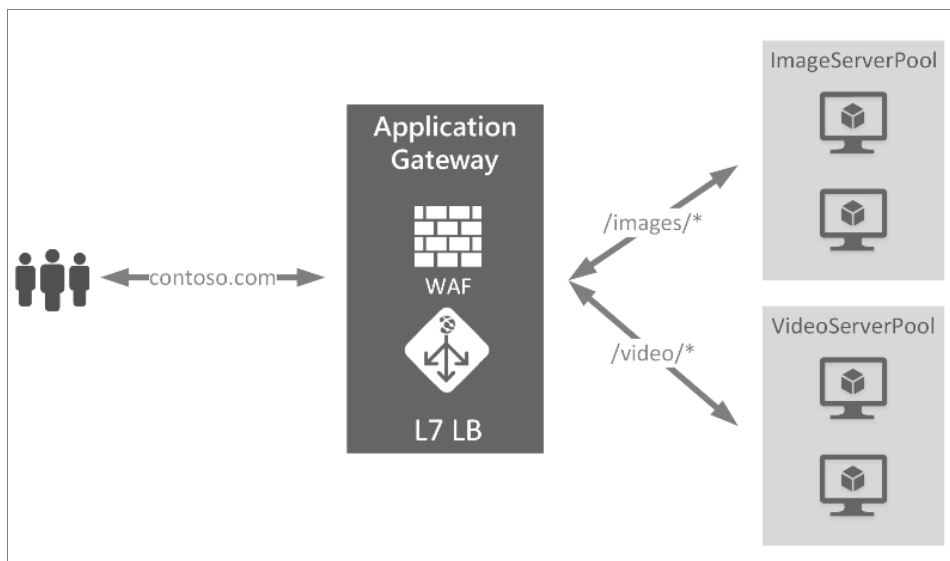


图 14.7-14

创建基于路径的规则如下。

基于 URL 路径的规则需要有自己的侦听器，在创建该规则之前，请确保有侦听器可用。

(1) 登录 Azure 门户，选择现有的应用程序网关，单击“规则”，如图 14.7-15 所示。



图 14.7-15

(2) 单击“基于路径”按钮，添加新的基于路径的规则，如图 14.7-16 所示。

- **名称**—这是基于路径的规则友好名称。
- **路径**—此设置可定义规则在转发流量时将要寻找的路径。
- **后端池**—此设置可定义要用于规则的后端。
- **HTTP 设置**—此设置可定义要用于规则的 HTTP 设置。



图 14.7-16



以上将基于路径的规则添加到现有应用程序网关，创建基于路径的规则后，即可对其进行编辑，以便轻松地添加其他规则。

**注意：**

路径：要匹配的路径模式列表。每个模式必须以 / 开头，“\*”只允许放在末尾处。有效示例包括 /xyz、/xyz\* 或 /xyz/\*。

## 14.8 使用 PowerShell 来备份 ARM 虚拟机

使用 PowerShell 可以自动化以下任务：

- (1) 创建恢复服务保管库；
- (2) 备份或保护 Azure VM；
- (3) 触发备份作业；
- (4) 监视备份作业；
- (5) 还原 Azure VM。

### 14.8.1 创建恢复服务保管库

以下步骤将引导用户创建恢复服务保管库。恢复服务保管库不同于备份保管库。

(1) 如果是首次使用 Azure 备份，则必须使用 Register-AzureRMResourceProvider cmdlet 将 Azure 恢复服务提供程序注册到订阅：

```
PS C:\> Register-AzureRmResourceProvider -ProviderNamespace "Microsoft.RecoveryServices "
```

(2) 恢复服务保管库是一种资源管理器资源，因此需要将它放在资源组中。你可以使用现有的资源组，也可以使用 New-AzureRmResourceGroup cmdlet 创建资源组。创建资源组时，请指定资源组的名称和位置：

```
PS C:\> New-AzureRmResourceGroup -Name "test-rg" -Location "China East "
```

(3) 使用 New-AzureRmRecoveryServicesVault cmdlet 创建保管库。确保为保管库指定的位置与用于资源组的位置是相同的：

```
PS C:\> New-AzureRmRecoveryServicesVault -Name "testvault" -ResourceGroup Name "test-rg" -Location "China East "
```

(4) 指定要使用的存储冗余类型；你可以使用本地冗余存储（LRS）或异地冗余存储（GRS）。以下示例显示，testVault 的 -BackupStorageRedundancy 选项设置为 GeoRedundant：

```
PS C:\> $vault1 = Get-AzureRmRecoveryServicesVault -Name "testVault"
PS C:\> Set-AzureRmRecoveryServicesBackupProperties -Vault $vault1 -BackupStorageRedundancy GeoRedundant
```

**注意：**

许多 Azure 备份 cmdlet 要求使用恢复服务保管库对象作为输入。因此，在变量中存储

备份恢复服务保管库对象可提供方便。

### 14.8.2 在订阅中查看保管库

使用 `Get-AzureRmRecoveryServicesVault` 查看当前订阅中所有保管库的列表。可以使用此命令来查看是否创建了新的保管库，或者查看订阅中的可用保管库。

运行 `Get-AzureRmRecoveryServicesVault` 命令即可列出订阅中的所有保管库。

```
PS C:\> Get-AzureRmRecoveryServicesVault
Name                : Contoso-vault
ID                  : /subscriptions/1234
Type                : Microsoft.RecoveryServices/vaults
Location            : ChinaEast
ResourceGroupName   : Contoso-docs-rg
SubscriptionId       : 1234-567f-8910-abc
Properties           : Microsoft.Azure.Commands.RecoveryServices.ARSVaultPr
                      operties
```

### 14.8.3 备份 Azure VM

既已经创建恢复服务保管库，可以使用它来保护虚拟机。但是，在应用保护之前，必须设置保管库上下文，并且需验证保护策略。保管库上下文定义了保管库中受保护的数据类型。保护策略是指对备份作业的运行时间以及每个备份快照的保留时长进行计划。

在 VM 上启用保护之前，必须设置保管库上下文。该上下文将应用到所有后续 `cmdlet`。

```
PS C:\> Get-AzureRmRecoveryServicesVault -Name testvault | Set-AzureRmRe
coveryServicesVaultContext
```

#### 14.8.3.1 创建保护策略

创建保管库时，它附带了一个默认策略。此策略会在每天的指定时间触发备份作业。根据默认策略，备份快照将保留 30 天。可以使用默认策略快速保护你的 VM，以后再使用不同的详细信息编辑该策略。

若要查看保管库中的可用策略列表，请使用 `Get-AzureRmRecoveryServicesBackupProtectionPolicy`：

```
PS C:\> Get-AzureRmRecoveryServicesBackupProtectionPolicy -WorkloadType
AzureVM
Name                WorkloadType      BackupManagementType BackupTime
DaysOfWeek
-----
DefaultPolicy       AzureVM          AzureVM              4/14/2016 5
:00:00 PM
```

NOTE:

PowerShell 中 `BackupTime` 字段的时区是 UTC。但是，在 Azure 门户预览中显示备份

时间时，时间将根据本地时区调整。

一个备份保护策略至少与一个保留策略相关联。保留策略定义在 Azure 备份中保留恢复点的时限。使用 `Get-AzureRmRecoveryServicesBackupRetentionPolicyObject` 可以查看默认保留策略。同理，可以使用 `Get-AzureRmRecoveryServicesBackupSchedulePolicyObject` 获取默认计划策略。计划和保留策略对象将用作 `New-AzureRmRecoveryServicesBackupProtectionPolicy cmdlet` 的输入。

备份保护策略定义对某个项目进行备份的时间和频率。`New-AzureRmRecoveryServicesBackupProtectionPolicy cmdlet` 创建用于保存备份策略信息的 PowerShell 对象。该备份策略用作 `Enable-AzureRmRecoveryServicesBackupProtection cmdlet` 的输入。

```
PS C:\> $schPol = Get-AzureRmRecoveryServicesBackupSchedulePolicyObject
-WorkloadType "AzureVM"
PS C:\> $retPol = Get-AzureRmRecoveryServicesBackupRetentionPolicyObject
-WorkloadType "AzureVM"
PS C:\> New-AzureRmRecoveryServicesBackupProtectionPolicy -Name
"NewPolicy" -WorkloadType AzureVM -RetentionPolicy $retPol -SchedulePolicy
$schPol
```

Name	WorkloadType	BackupManagementType	BackupTime
DaysOfWeek			
-----	-----	-----	-----
-----			
NewPolicy	AzureVM	AzureVM	4/24/2016 1:30:00 AM

#### 14.8.3.2 启用保护

启用保护涉及两个对象 - 项和策略。需要提供这两个对象才能为保管库启用保护。将策略与保管库关联之后，将在策略计划中定义的时间触发备份工作流。

在非加密型资源管理器 VM 上启用保护：

```
PS C:\> $pol=Get-AzureRmRecoveryServicesBackupProtectionPolicy -Name
"NewPolicy"
PS C:\> Enable-AzureRmRecoveryServicesBackupProtection -Policy $pol -Name
"V2VM" -ResourceGroupName "RGName1"
```

若要在加密型 VM 上启用保护[使用 BEK 和 KEK 加密]，需提供相应权限，允许 Azure 备份服务读取密钥保管库中的密钥和机密：

```
PS C:\> Set-AzureRmKeyVaultAccessPolicy -VaultName 'KeyVaultName'
-ResourceGroupName 'RGNameOfKeyVault' -PermissionsToKeys backup,get,list
-PermissionsToSecrets get,list -ServicePrincipalName
262044b1-e2ce-469f-a196-69ab7ada62d3
PS C:\> $pol=Get-AzureRmRecoveryServicesBackupProtectionPolicy -Name
"NewPolicy"
PS C:\> Enable-AzureRmRecoveryServicesBackupProtection -Policy $pol -Name
"V2VM" -ResourceGroupName "RGName1"
```

**注意：**

如果使用 Azure Government 云，则对 Set-azurermkeyvaultaccesspolicy cmdlet 中的参数 -ServicePrincipalName 使用值 ff281ffe-705c-4f53-9f37-a40e6f2c68f3。

## 14.4.3.3 针对经典 VM

```
PS C:\> $pol=Get-AzureRmRecoveryServicesBackupProtectionPolicy -Name
"NewPolicy"
PS C:\> Enable-AzureRmRecoveryServicesBackupProtection -Policy $pol -Name
"V1VM" -ServiceName "ServiceName1"
```

## 14.8.3.4 修改保护策略

若要修改策略，请修改 BackupSchedulePolicyObject 或 BackupRetentionPolicy 对象，并使用 Set-AzureRmRecoveryServicesBackupProtectionPolicy 修改策略。

以下示例将保留计数更改为 365。

```
PS C:\> $retPol = Get-AzureRmRecoveryServicesBackupRetentionPolicyObject
-WorkloadType "AzureVM"
PS C:\> $retPol.DailySchedule.DurationCountInDays = 365
PS C:\> $pol= Get-AzureRmRecoveryServicesBackupProtectionPolicy -Name
NewPolicy
PS C:\> Set-AzureRmRecoveryServicesBackupProtectionPolicy -Policy $pol
-RetentionPolicy $RetPol
```

## 14.8.4 运行初始备份

在首次备份项时，备份计划会触发完整备份。对于后续备份，备份形式为增量复制。若要强制初始备份在某个时间发生甚至立刻发生，可以使用 Backup-AzureRmRecoveryServicesBackupItem cmdlet：

```
PS C:\> $namedContainer = Get-AzureRmRecoveryServicesBackupContainer -Co
ntainerType "AzureVM" -Status "Registered" -FriendlyName 'V2VM'
PS C:\> $item = Get-AzureRmRecoveryServicesBackupItem -Container $namedC
ontainer -WorkloadType "AzureVM"
PS C:\> $job = Backup-AzureRmRecoveryServicesBackupItem -Item $item
```

WorkloadName	Operation	Status	StartTime
EndTime		JobID	
V2VM	Backup	InProgress	4/23/2016 5:00:30 PM
		cf4b3ef5-2fac-4c8e-a215-d2eba4124f27	

**NOTE:**

PowerShell 中 StartTime 和 EndTime 字段的时区为 UTC。

### 14.8.5 监视备份作业

在 Azure 备份中，大多数长时间运行的操作都是作为作业来建模的。

若要获取正在进行的作业的最新状态，请使用 `Get-AzureRmRecoveryservicesBackupJob` cmdlet。

```
PS C:\> $joblist = Get-AzureRmRecoveryservicesBackupJob -Status InProgress
PS C:\> $joblist[0]
```

WorkloadName	Operation	Status	StartTime	EndTime	JobID
V2VM	Backup	InProgress		4/23/2016 5:00:30 PM	cf4b3ef5-2fac-4c8e-a215-d2eba4124f27

与其使用额外的不必要的代码来轮询这些作业的完成情况，不如使用 `Wait-AzureRmRecoveryServicesBackupJob` cmdlet。该 cmdlet 暂停操作的执行，直到作业完成或达到了指定的超时值。

```
PS C:\> Wait-AzureRmRecoveryServicesBackupJob -Job $joblist[0] -Timeout 43200
```

### 14.8.6 还原 Azure VM

如果使用 PowerShell，从恢复点创建磁盘和配置信息即可完成还原操作。还原操作不会创建虚拟机。我们提供了从磁盘创建虚拟机的说明。但是，若要完全还原 VM，需要完成以下步骤：

- 选择 VM；
- 选择恢复点；
- 还原磁盘；
- 通过存储磁盘创建 VM。

下图显示了从 `RecoveryServicesVault` 到 `BackupRecoveryPoint` 的对象层次结构，如图 14.8-1 所示。

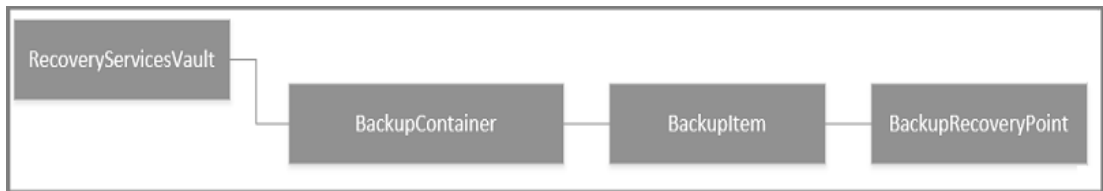


图 14.8-1

若要还原备份数据，请确定已备份项目以及保留了时间点数据的恢复点。然后，请使用 `Restore-AzureRmRecoveryServicesBackupItem` cmdlet 将数据从保管库还原到客户的账户。

### 14.8.6.1 选择 VM

若要获取用于标识正确备份项的 PowerShell 对象，请从保管库中的容器开始，按对象层次结构进行操作。若要选择代表 VM 的容器，请使用 `Get-AzureRmRecoveryServicesBackupContainer` cmdlet，然后通过管道将其传递给 `Get-AzureRmRecoveryServicesBackupItem` cmdlet。

```
PS C:\> $namedContainer = Get-AzureRmRecoveryServicesBackupContainer -
ContainerType AzureVM -Status Registered -FriendlyName 'V2VM'
PS C:\> $backupitem = Get-AzureRmRecoveryServicesBackupItem -Container $
namedContainer -WorkloadType " AzureVM "
```

### 14.8.6.2 选择恢复点

使用 `Get-AzureRmRecoveryServicesBackupRecoveryPoint` cmdlet 列出备份项的所有恢复点。然后选择要还原的恢复点。如果不确定要使用的恢复点，最好选择列表中最新的 `RecoveryPointType = AppConsistent` 恢复点。

在以下脚本中，变量 `$rp` 是一个数组，其中包含所选备份项的恢复点。该数组按时间进行反向排序，以最新的恢复点作为索引 0。使用标准 PowerShell 数组索引选取恢复点。例如：`$rp[0]` 将选择最新的恢复点。

```
PS C:\> $startDate = (Get-Date).AddDays(-7)
PS C:\> $endDate = Get-Date
PS C:\> $rp = Get-AzureRmRecoveryServicesBackupRecoveryPoint -Item $back
upitem -StartDate $startdate.ToUniversalTime() -EndDate $enddate.ToUniversa
lTime()
PS C:\> $rp[0]
RecoveryPointAdditionalInfo :
SourceVMStorageType        : NormalStorage
Name                        : 15260861925810
ItemName                    : VM;iaasvmcontainer;RGName1;V2VM
RecoveryPointId             : /subscriptions/XX/resourceGroups/ RGName1/
providers/Microsoft.RecoveryServices/vaults/testvault/backupFabrics/Azure/p
rotectionContainers/IaasVMContainer;iaasvmcontainer;RGName1;V2VM/protectedI
tems/VM;iaasvmcontainer; RGName1;V2VM
                             /recoveryPoints/15260861925810
RecoveryPointType           : AppConsistent
RecoveryPointTime           : 4/23/2016 5:02:04 PM
WorkloadType                : AzureVM
ContainerName               : IaasVMContainer;iaasvmcontainer; RGName1;V
2VM
ContainerType               : AzureVM
BackupManagementType        : AzureVM
```

### 14.8.6.3 还原磁盘

使用 `Restore-AzureRmRecoveryServicesBackupItem` cmdlet 将备份项的数据和配置还原

到某个恢复点。确定某个恢复点后，即可使用它作为 **-RecoveryPoint** 参数的值。在以前的示例代码中，选择了 `$rp[0]` 作为要使用的恢复点。在下面的示例代码中，指定了 `$rp[0]` 作为还原到磁盘时要使用的恢复点。

还原磁盘和配置信息：

```
PS C:\> $restorejob = Restore-AzureRmRecoveryServicesBackupItem -RecoveryPoint $rp[0] -StorageAccountName DestAccount -StorageAccountResourceGroupName DestRG
PS C:\> $restorejob
```

WorkloadName	Operation	Status	StartTime
V2VM	Restore	InProgress	4/23/2016 5:00:30 PM

```
-----
JobID
-----
cf4b3ef5-2fac-4c8e-a215-d2eba4124f27
```

可以使用 **Wait-AzureRmRecoveryServicesBackupJob** 等待还原作业完成：

```
PS C:\> Wait-AzureRmRecoveryServicesBackupJob -Job $restorejob -Timeout 43200
```

还原作业完成后，可以使用 **Get-AzureRmRecoveryServicesBackupJobDetails** cmdlet 获取还原操作的详细信息。`JobDetails` 属性提供重建 VM 所需的信息：

```
PS C:\> $restorejob = Get-AzureRmRecoveryServicesBackupJob -Job $restorejob
PS C:\> $details = Get-AzureRmRecoveryServicesBackupJobDetails -Job $restorejob
```

还原磁盘以后，即可转到下一部分以了解如何创建 VM。

### 14.8.7 从还原的磁盘创建 VM

还原磁盘以后，即可通过以下步骤从磁盘创建和配置虚拟机。

如果使用还原的磁盘创建加密型 VM，用户角色应有权执行 **Microsoft.KeyVault/vaults/deploy/action**。如果用户角色不具有此权限，请创建具有此操作的自定义角色。请参阅 **Azure RBAC** 中的自定义角色，了解更多详细信息。

(1) 查询已还原磁盘属性以获取作业详细信息：

```
PS C:\> $properties = $details.properties
PS C:\> $storageAccountName = $properties["Target Storage Account Name"]
PS C:\> $containerName = $properties["Config Blob Container Name"]
PS C:\> $blobName = $properties["Config Blob Name"]
```

(2) 设置 Azure 存储上下文和还原 JSON 配置文件：

```
PS C:\> Set-AzureRmCurrentStorageAccount -Name $storageaccountname -ResourceGroupName testvault
PS C:\> $destination_path = "C:\vmconfig.json"
```

```
PS C:\> Get-AzureStorageBlobContent -Container $containerName -Blob $blobName -Destination $destination_path
PS C:\> $obj = ((Get-Content -Path $destination_path -Raw -Encoding Unicode)).TrimEnd([char]0x00) | ConvertFrom-Json
```

(3) 使用 JSON 配置文件来创建 VM 配置。

```
PS C:\> $vm = New-AzureRmVMConfig -VMSize $obj.HardwareProfile.VirtualMachineSize -VMName "testrestore"
```

(4) 附加 OS 磁盘和数据磁盘。

对于非加密型 VM:

```
PS C:\> Set-AzureRmVMOSDisk -VM $vm -Name "osdisk" -VhdUri $obj.StorageProfile.OSDisk.VirtualHardDisk.Uri -CreateOption "Attach"
PS C:\> $vm.StorageProfile.OsDisk.OsType = $obj.StorageProfile.OSDisk.OperatingSystemType
PS C:\> foreach($dd in $obj.StorageProfile.DataDisks)
{
    $vm = Add-AzureRmVMDataDisk -VM $vm -Name "datadisk1" -VhdUri $dd.VirtualHardDisk.Uri -DiskSizeInGB 127 -Lun $dd.Lun -CreateOption Attach
}
```

对于加密型 VM，需在附加磁盘前指定密钥保管库信息：

```
PS C:\> Set-AzureRmVMOSDisk -VM $vm -Name "osdisk" -VhdUri $obj.StorageProfile.OSDisk.VirtualHardDisk.Uri -DiskEncryptionKeyUrl "https://ContosoKeyVault.vault.azure.cn:443/secrets/ContosoSecret007" -DiskEncryptionKeyVaultId "/subscriptions/abcdedf007-4xyz-1a2b-0000-12a2b345675c/resourceGroups/ContosoRG108/providers/Microsoft.KeyVault/vaults/ContosoKeyVault" -KeyEncryptionKeyUrl "https://ContosoKeyVault.vault.azure.cn:443/keys/ContosoKey007" -KeyEncryptionKeyVaultId "/subscriptions/abcdedf007-4xyz-1a2b-0000-12a2b345675c/resourceGroups/ContosoRG108/providers/Microsoft.KeyVault/vaults/ContosoKeyVault" -CreateOption "Attach" -Windows
PS C:\> $vm.StorageProfile.OsDisk.OsType = $obj.StorageProfile.OSDisk.OperatingSystemType
PS C:\> foreach($dd in $obj.StorageProfile.DataDisks)
{
    $vm = Add-AzureRmVMDataDisk -VM $vm -Name "datadisk1" -VhdUri $dd.VirtualHardDisk.Uri -DiskSizeInGB 127 -Lun $dd.Lun -CreateOption Attach
}
```

(5) 设置网络设置：

```
PS C:\> $nicName = "p1234"
PS C:\> $pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName "test" -Location "ChinaEast" -AllocationMethod Dynamic
PS C:\> $vnet = Get-AzureRmVirtualNetwork -Name "testvNET"
```



```
-ResourceGroupName " test "
PS C:\> $nic = New-AzureRmNetworkInterface -Name $nicName
-ResourceGroupName " test " -Location " ChinaEast " -SubnetId $vnet.Subnets
[$subnetindex] .Id -PublicIpAddressId $pip.Id
PS C:\> $vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

(6) 创建虚拟机:

```
PS C:\> New-AzureRmVM -ResourceGroupName " test " -Location " ChinaEast "
-VM $vm
```

## 14.9 面向 Internet 的负载均衡

面向 Internet 的负载均衡器是一种第四层 (TCP、UDP) 类型的负载均衡器, 其将传入流量的公用 IP 地址和端口号映射到虚拟机的专用 IP 地址和端口号, 对于来自虚拟机的响应流量, 则进行反向的映射。借助负载均衡规则, 可在多个虚拟机或服务之间分配特定类型的流量。例如, 可将 Web 请求流量负载分配到多个 Web 服务器或 Web 角色。

下图显示了公用和专用 TCP 端口 80/443 的 Web 流量的负载均衡终结点, 由三个虚拟机共享。这三个虚拟机位于一个负载均衡集中, 如图 14.9-1 所示。

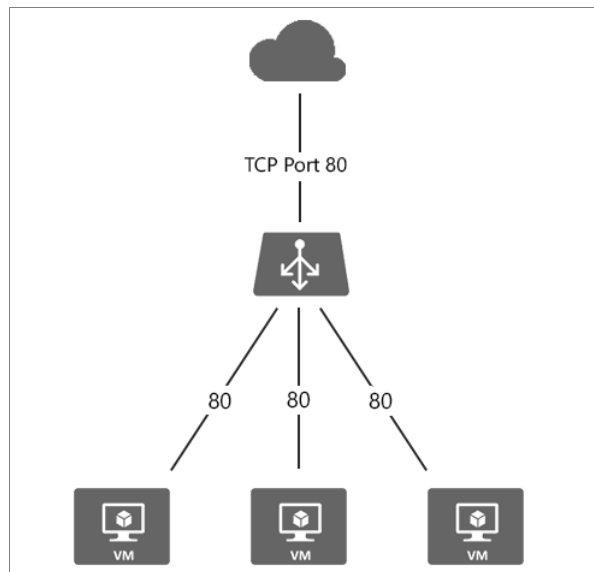


图 14.9-1

当 Internet 客户端将网页请求发送到 TCP 端口 80/443 上的公共 IP 地址时, Azure Load Balancer 会在负载均衡集中的三个虚拟机之间分发请求。

默认情况下, Azure Load Balancer 在多个虚拟机实例之间平均分发网络流量。另外, 还可以配置会话关联。

### 14.9.1 创建面向 Internet 的负载均衡器

需要实现的目标（资源管理器部署模型）：

- 在端口 80 上创建一个接收网络流量的负载均衡器，并将负载均衡流量发送到虚拟机 “lytWeb1” 和 “lytWeb2”；
- 在负载均衡器后面创建虚拟机的远程桌面访问/SSH 的 NAT 规则；
- 创建运行状况探测，如图 14.9-2 所示。

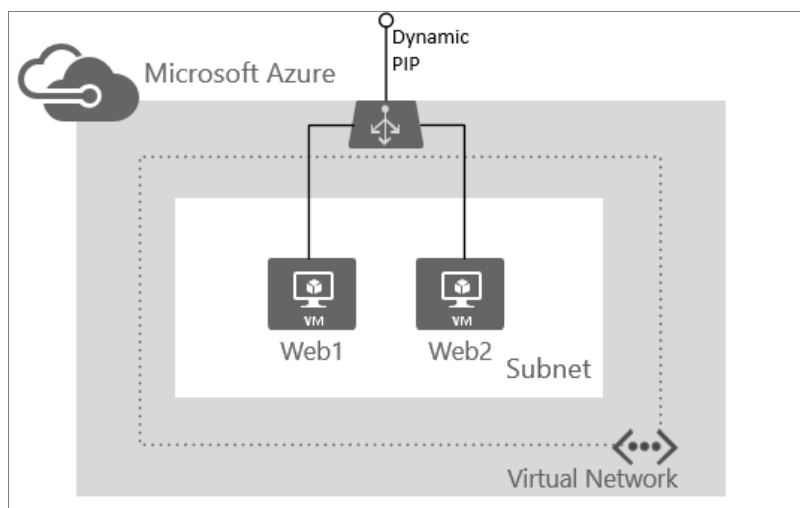


图 14.9-2

前期配置准备：

- 前端 IP 配置—包含传入网络流量的公共 IP 地址。
- 后端地址池—包含从负载均衡器接收网络流量的虚拟机网络接口（NIC）。
- 负载均衡规则—包含将负载均衡器上的公共端口映射到后端地址池中的端口的规则。
- 入站 NAT 规则—包含将负载均衡器上的公共端口映射到后端地址池中特定虚拟机的端口的规则。
- 探测器—包含用于检查后端地址池中虚拟机实例的可用性的运行状况探测器。

注意：

本示例假定已有名为 lytvnet 的虚拟网络，lytvnet 内有一个名为 lytsubnet 的子网以及两个分别名为 lytWeb1 和 lytWeb2 的 VM，这两个 VM 都位于 lytvnet 中名为 lytAvailSet 的可用性集中。

#### 14.9.1.1 在 Azure 门户预览中设置负载均衡器

(1) 从浏览器导航到 Azure 门户预览：<http://portal.azure.cn>，并使用你的 Azure 账户登录。

- (2) 在屏幕的左上方, 选择“新建”>“网络”>“负载均衡器”。
- (3) 在“创建负载均衡器”边栏选项卡中, 为负载均衡器键入一个名称。此例中使用名称 lytLoadBalancer。
- (4) 在“类型”下, 选择“公共”。
- (5) 在“公共 IP 地址”下, 创建名为 lytPublicIP 的新公共 IP。
- (6) 在“资源组”下, 选择“lytrg”。然后, 选择相应的位置, 并单击“确定”。然后, 负载均衡器将开始部署, 成功完成部署需要几分钟的时间, 如图 14.9-3 所示。



创建负载均衡器

\* 名称  
lytLoadBalancer ✓

\* 类型 ⓘ  
☒ 公共 ☐ 内部

\* 公共 IP 地址  
(新) lytPublicIP >

\* 订阅  
WATSTest03 ▼

\* 资源组 ⓘ  
☐ 新建 ☒ 使用现有项  
lytrg ▼

\* 位置  
中国北部 ▼

图 14.9-3

#### 14.9.1.2 创建后端地址池

- (1) 成功部署负载均衡器后, 请从资源中选择它。在“设置”下, 选择“后端池”。为后端池键入名称。然后单击显示的边栏选项卡顶部附近的“添加”按钮。
- (2) 在“添加后端池”边栏选项卡中, 单击“添加虚拟机”。在“可用性集”下, 选择“选择可用性集”, 然后选择“lytAvailSet”。接下来, 在边栏选项卡的“虚拟机”部分下选择“选择虚拟机”, 然后单击为实现负载均衡而创建的两个 VM“Web1”和“Web2”。请确保这两个 VM 左侧都带有蓝色复选标记。然后, 依次单击该边栏选项卡中的“选择”、“选择虚拟机”边栏选项卡中的“确定”和“添加后端池”边栏选项卡中的“确定”。如图 14.9-4 所示。

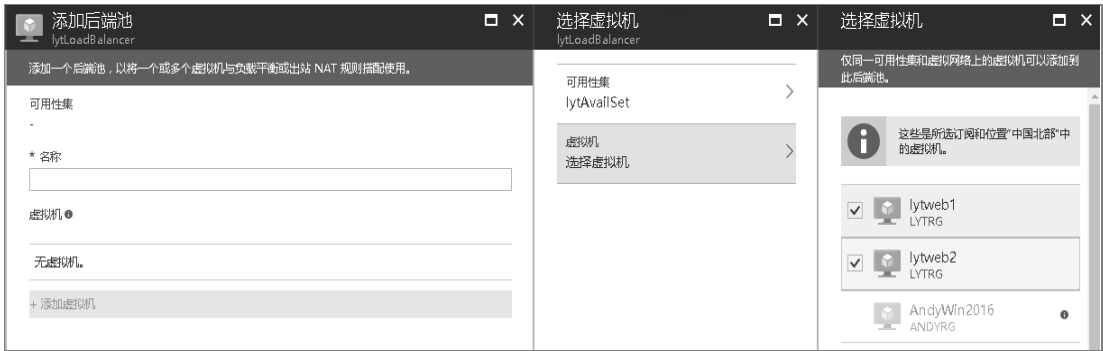


图 14.9-4

(3) 进行相关检查，确保通知下拉列表中除了针对 VM lytWeb1 和 lytWeb2 的网络接口更新外，还包含有关保存负载均衡器后端池的更新。

### 14.9.1.3 创建探测器、LB 规则和 NAT 规则

#### 1. 创建运行状况探测器

在负载均衡器的“设置”下，选择“探测器”。然后，单击边栏选项卡顶部的“添加”。

可通过两种方法配置探测器：HTTP 或 TCP。此示例演示 HTTP 的配置，但可以按类似的方法配置 TCP。更新必要的信息。如前文所述，lytLoadBalancer 将在端口 80 上实现流量的负载均衡。所选路径为 HealthProbe.aspx，时间间隔为 15 秒，不正常阈值为 2。完成后，单击“确定”以创建探测器。

将指针悬停在“i”图标上可深入了解这些单个配置，以及如何更改它们以满足特定要求，如图 14.9-5 所示。



图 14.9-6

## 2. 创建负载均衡规则

单击负载均衡器的“设置”部分中的负载均衡规则。在“新建”边栏选项卡中，单击“添加”。为规则命名。此处使用 HTTP。选择前端和后端端口。此处，两个端口均选择 80。选择“lytBackendPool”作为后端池，选择以前创建的“HealthProbe”作为探测器。可根据用户的需求设置其他配置。然后，单击“确定”保存负载均衡规则，如图 14.9-7 所示。



图 14.9-7

## 3. 创建入站 NAT 规则

单击负载均衡器的“设置”部分下的“入站 NAT 规则”。在“新建”边栏选项卡中，单击“添加”。然后，为入站 NAT 规则命名。此处使用名称 InboundNATrule1。目标应是以前创建的公共 IP。在“服务”下选择“自定义”，并选择要使用的协议。此处选择 TCP。输入端口（1125）和目标端口（此例中为 3389）。然后单击“确定”保存此规则。

创建第一条规则后，对名为 InboundNATrule2 的第二个入站 NAT 规则重复此步骤（从端口 1126 到目标端口 3389），如图 14.9-8 所示。

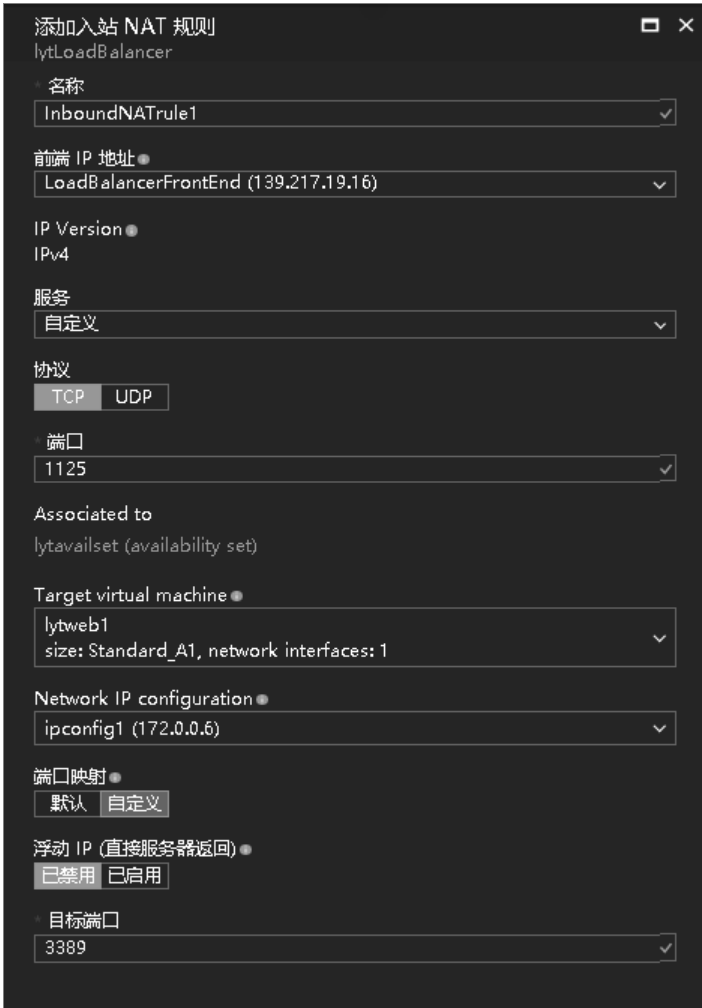


图 14.9-8

## 14.9.2 案例介绍

### 14.9.2.1 案例一

A 公司目前有一个 Web Site, 外网出口通过 HAPROXY 对后端 2 台 Web Server 基于四层流量实现软负载均衡, 但是随着业务量的发展及用户访问量的迅猛增长, 当前的负载均衡器及 Web 服务器均已达到性能瓶颈, 已经严重影响到用户体验。此时, 信息化部门考虑到流程申请复杂、采购周期长、硬件设备及运维成本也将大幅增加。从公司长远战略层面考虑, 最终决定将 Web 服务整体迁移至 Microsoft Azure 云平台, 并实现快速上线。

部署方案:

前端采用面向 Internet 的负载均衡器, 后端使用 3 台 Azure VM(DS11)作为 Web Server, 实现对 Internet 访问量基于 level4 的 HTTP 负载均衡目标。

操作步骤：（采用 PowerShell 实现，以 2 台 VM 为例）

## 1. 将 PowerShell 设置为使用 Resource Manager 模式

（1）登录 Azure（如有多个订阅，需指定当前的订阅）

```
Login-AzureRmAccount -EnvironmentName AzureChinaCloud
```

（2）创建资源组（若要使用现有资源组，请跳过此步骤）

```
New-AzureRmResourceGroup -Name NRP-RG -location "China East "
```

## 2. 为前端 IP 池创建虚拟网络和公共 IP 地址

（1）创建虚拟网络和子网，如图 14.9-9 所示。

```
$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name LB-Subnet
-AddressPrefix 11.0.0.0/24
New-AzureRmvirtualNetwork -Name LYTVNet -ResourceGroupName LYT-RG -Location
'China East' -AddressPrefix 11.0.0.0/16 -Subnet $backendSubnet
```

```
PS C:\Users\luyitong> $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name LB-Subnet -AddressPrefix 11.0.0.0/24
PS C:\Users\luyitong> New-AzureRmvirtualNetwork -Name LYTVNet -ResourceGroupName LYT-RG -Location 'China East' -AddressP
refix 11.0.0.0/16 -Subnet $backendSubnet
警告: The output object type of this cmdlet will be modified in a future release.

Name                : LYTVNet
ResourceGroupName   : LYT-RG
Location            : chinaeast
Id                  : /subscriptions/5dda9164-06db-4412-ad53-bb0521e5e1/resourceGroups/LYT-RG/providers/Microsoft.
Network/virtualNetworks/LYTVNet
Etag                : W/"cf48d472-8f45-44fe-9617-1eba8c2af262"
ResourceGuid        : 109b9bd8-809d-4a05-9415-f58f18f72b8f
ProvisioningState    : Succeeded
Tags                :
AddressSpace        : {
                        "AddressPrefixes": [
                          "11.0.0.0/16"
                        ]
                      }
DhcpOptions          : 0
Subnets             : [
                        {
                          "Name": "LB-Subnet",
                          "Etag": "W/"cf48d472-8f45-44fe-9617-1eba8c2af262\"",
                          "Id": "/subscriptions/5dda9164-06db-4412-ad53-bb0521e5e1/resourceGroups/LYT-RG/providers
/Microsoft.Network/virtualNetworks/LYTVNet/subnets/LB-Subnet",
                          "AddressPrefix": "11.0.0.0/24",
                          "IpConfigurations": [],
                          "ResourceNavigationLinks": [],
                          "ProvisioningState": "Succeeded"
                        }
                      ]
VirtualNetworkPeerings : []
```

图 14.9-9

（2）使用 DNS 名称 loadbalancernrp.chinaeast.chinacloudapp.cn 创建要由前端 IP 池使用的名为 PublicIP 的 Azure 公共 IP 地址资源，以下命令使用静态分配类型

```
$publicIP = New-AzureRmPublicIpAddress -Name PublicIP -ResourceGroupName
LYT-RG -Location 'China East' -AllocationMethod Static -DomainNameLabel
loadbalancerlyt
```

注意：

负载均衡器将公共 IP 的域标签用作其 FQDN 的前缀。这与经典部署模型不同，后者使用云

服务作为负载均衡器 FQDN。在此示例中，FQDN 是 loadbalancerlyt.chinanorth.chinacloudapp.cn。

### 3. 创建前端 IP 池和后端地址池

#### (1) 创建使用 PublicIP 资源且名为 LB-Frontend 的前端 IP 池

```
$frontendIP = New-AzureRmLoadBalancerFrontendIpConfig -Name LB-Frontend
-PublicIpAddress $publicIP
```

#### (2) 创建名为 LB-backend 的后端地址池

```
$backendpool = New-AzureRmLoadBalancerBackendAddressPoolConfig -Name
LB-backend
```

### 4. 创建 NAT 规则、负载均衡器规则、探测器和负载均衡器

#### (1) 创建 NAT 规则

如果 VM 只配置一个 NIC，这个 NIC 不能同时既用于 NAT 又绑定 Public IP，所以只能取其一。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name RDP1
-FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3335
-BackendPort 3389
$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name RDP2
-FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3336
-BackendPort 3389
```

#### (2) 创建运行状况探测器（有两种方法可以配置探测器：）

##### (a) HTTP 探测器：

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe
-RequestPath 'HealthProbe.aspx' -Protocol http -Port 80 -IntervalInSeconds 15
-ProbeCount 2
```

##### (b) TCP 探测器：

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe
-Protocol Tcp -Port 80 -IntervalInSeconds 15 -ProbeCount 2
```

#### (3) 创建负载均衡器规则

```
$lbrule=New-AzureRmLoadBalancerRuleConfig -Name HTTP -FrontendIpConfiguration
$frontendIP -BackendAddressPool $backendpool -Probe $healthProbe -Protocol Tcp
-FrontendPort 80 -BackendPort 80
```

#### (4) 使用之前创建的对象创建负载均衡器

```
$LYTLB = New-AzureRmLoadBalancer -ResourceGroupName LYT-RG -Name LYT-LB
-Location 'China East' -FrontendIpConfiguration $frontendIP -LoadBalancingRule
$lbrule -BackendAddressPool $backendpool -Probe $healthProbe
```



## 5. 创建 NIC

创建网络接口（或修改现有接口），并将其关联到负载均衡器规则和探测器

(1) 获取需创建 NIC 的虚拟网络和虚拟网络子网：

```
$vnet = Get-AzureRmVirtualNetwork -Name LYTNet -ResourceGroupName LYT-RG
$backendSubnet = Get-AzureRmVirtualNetworkSubnetConfig -Name LB-Subnet
-VirtualNetwork $vnet
```

(2) 创建名为 lb-nic1 的 NIC，并将其与第一个（且仅有的）后端地址池相关联：

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName LYT-RG -Name
lb-nic1 -Location 'China East' -PrivateIpAddress 11.0.0.4 -Subnet
$backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0]
```

(3) 创建名为 lb-nic2 的 NIC，并将其与第一个（且仅有的）后端地址池相关联：

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName LYT-RG -Name
lb-nic1 -Location 'China East' -PrivateIpAddress 11.0.0.4 -Subnet
$backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0]
```

(4) 使用 Add-AzureRmVMNetworkInterface cmdlet 将两个 NIC 分配给 2 台 VM（此步骤在创建虚拟机的环节执行）。

## 6. 创建虚拟机

创建 VM 请参考官方文档：

<https://www.azure.cn/documentation/>（Azure 官方文档中心）

## 7. 向负载均衡器添加网络接口

(1) 检索 Azure 中的负载均衡器

将负载均衡器资源加载到变量中（如果你还没有这样做）。该变量称为 \$lb。使用与先前创建的负载均衡器资源相同的名称：

```
$lb= get-azurermloadbalancer -name LYT-LB -resourcegroupname LYT-RG
```

(2) 将后端配置加载到变量：

```
$backend=Get-AzureRmLoadBalancerBackendAddressPoolConfig -name LB-backend
-LoadBalancer $lb
```

(3) 将创建好的网络接口加载到变量中。变量名为 \$nic（lb-nic2 略）：

```
$nic =get-azurermmnetworkinterface -name lb-nic1 -resourcegroupname LYT-RG
```

(4) 更改网络接口上的后端配置：

```
$nic.IpConfigurations[0].LoadBalancerBackendAddressPools=$backend
```

(5) 保存网络接口对象：

```
Set-AzureRmNetworkInterface -NetworkInterface $nic
```

将网络接口添加到负载均衡器后端池后，它会根据该负载均衡器资源的负载均衡规则开始接收网络流量

(6) 查看所创建的 LoadBalancer 的详细信息：

```
Get-AzureRmLoadBalancer -Name LYT-LB -resourcegroupname LYT-RG
```

## 8. 删除负载均衡器

使用命令 `Remove-AzureLoadBalancer` 删除之前在 LYT-RG 资源组中创建的 LYT-LB 负载均衡器：

```
Remove-AzureRmLoadBalancer -Name LYT-LB -ResourceGroupName LYT-RG
```

### 14.9.2.2 案例二

B 公司使用了 Azure 云服务，通过 Azure 负载均衡器实现媒体端点的负载，由于在媒体上传等特殊场景下，实际的数据上传是通过 UDP 实现，而控制层则通过 TCP 实现，客户端先向负载均衡公共地址发起 TCP 会话，然后被导向至特定 DIP，这个通道将保持活动状态，用户监控连接状态。而同一个客户端计算机向同一个负载均衡公共端点发起新的 UDP 会话，此时希望这个连接也被导向至与前面 TCP 连接相同的 DIP 端点，使媒体上传能够以高吞吐量执行的同时也能够维持 TCP 控制通道。

**解决方案：**将负载均衡器当前默认的分发模式由 5 元组改为 2 元组。

我们引入了一种称为源 IP 关联（也称为会话关联或客户端 IP 关联）的新分发模式。Azure 负载均衡器可以配置为使用 2 元组（源 IP、目标 IP）或 3 元组（源 IP、目标 IP、协议）将流量映射到可用服务器上。使用源 IP 关联，同一客户端计算机上发起的连接都会转到同一个 DIP 端点，如图 14.9-10 所示。

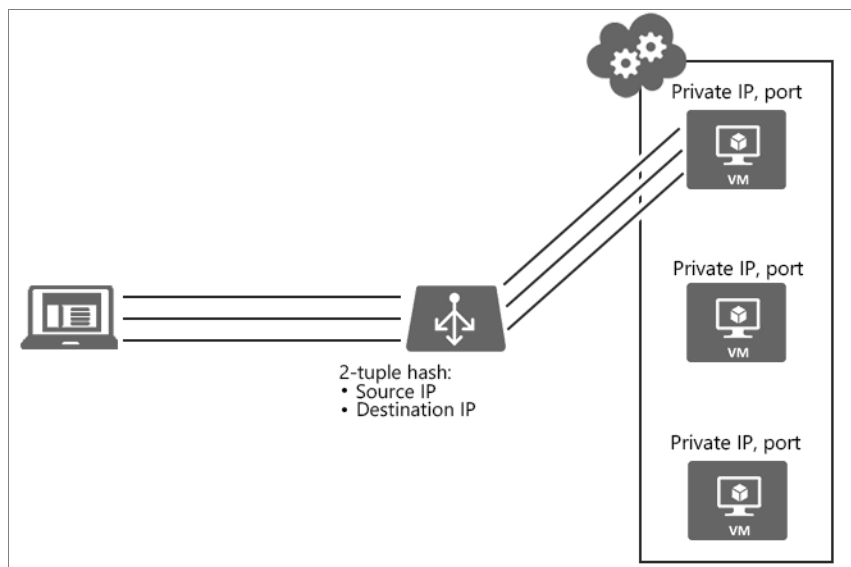


图 14.9-10

配置步骤:

(1) 修改负载均衡器的分发模式:

```
Set-AzureLoadBalancedEndpoint -ServiceName "XXX" -LBSetName http -Protocol
tcp -LocalPort 80 -ProbeProtocol TCP -ProbePort 80 -LoadBalancerDistribution
sourceIP
```

(2) 查看当前端点负载均衡器分发模式, 已修改成 sourceIP, 即 2 元组:

```
Get-AzureVM -ServiceName "XXX" -Name "XX" | Get-AzureEndpoint
```

输出结果:

```
VERBOSE:8:25:52 PM -CompletedOperation:GetDeployment
LBSetName :MyLoadBalancedSet
LocalPort :80
Name :HTTP
Port :80
Protocol : tcp
Vip :65.52.xxx.xxx
ProbePath :
ProbePort :80
ProbeProtocol : tcp
ProbeIntervalInSeconds :15
ProbeTimeoutInSeconds :31
EnableDirectServerReturn :False
Acl :{}
InternalLoadBalancerName :
IdleTimeoutInMinutes :15
LoadBalancerDistribution : sourceIP
```

注意:

LoadBalancerDistribution 可以设置为 sourceIP (用于 2 元组 (源 IP、目标 IP) 负载均衡)、sourceIPProtocol (用于 3 元组 (源 IP、目标 IP、协议) 负载均衡) 或 none (如果想要使用 5 元组负载均衡的默认行为)。

#### Autoscale\VMSS

虚拟机规模集是一种 Azure 计算资源, 可用于部署和管理一组相同的 VM。VM 规模集中的所有 VM 均采用相同的配置, 而无需对 VM 进行预配, 这可更简便地生成面向大计算、大数据、容器化工作负荷的大规模服务。

对于需要扩大和缩小计算资源的应用程序, 缩放操作在容错域和更新域之间进行隐式平衡。虚拟机规模集可让你以集的形式管理多个 VM。概括而言, 规模集具有以下特点:

- 几分钟内创建上百个相同的虚拟机;
- 快速扩展你的大计算和大数据应用程序;
- 依赖于集成负载均衡和自动扩展;
- 高可用性。每个规模集将它的 VM 放入具有 5 个容错域(FD)和 5 个更新域(UD)的可用性集, 以确保可用性;

- 简化 VM 的部署、管理和清理；
- 支持常用的 Windows 和 Linux 版本以及自定义映像。

### 14.10.1 创建和管理 VM 规模集

可以在 Azure 预览中选择“新建”，然后在搜索栏中键入“规模”，来创建 VM 规模集。结果中会看到“虚拟机规模集”。从这里，可以填写必填字段，自定义和部署规模集。请注意，门户中也提供了基于 CPU 使用情况设置自动调整规模的基本规则的选项。

也可以使用 JSON 模板和 REST API 定义和部署 VM 规模集，就像定义和部署单个 Azure Resource Manager VM 一样。因此，可以使用任何标准的 Azure 资源管理器部署方法。

(1) 首先，在 Web 浏览器中登录到 Azure 门户预览。在“新建”栏下方选择“虚拟机规模集”条目。

(2) 使用默认设置并快速创建规模集：

- 在基本边栏选项卡上，输入规模集名称；
- 选择所需的 OS 类型，输入用户名、密码，选择使用的“订阅”；
- 输入所需的资源组名称和位置，然后单击 OK，如图 14.10-1 所示。

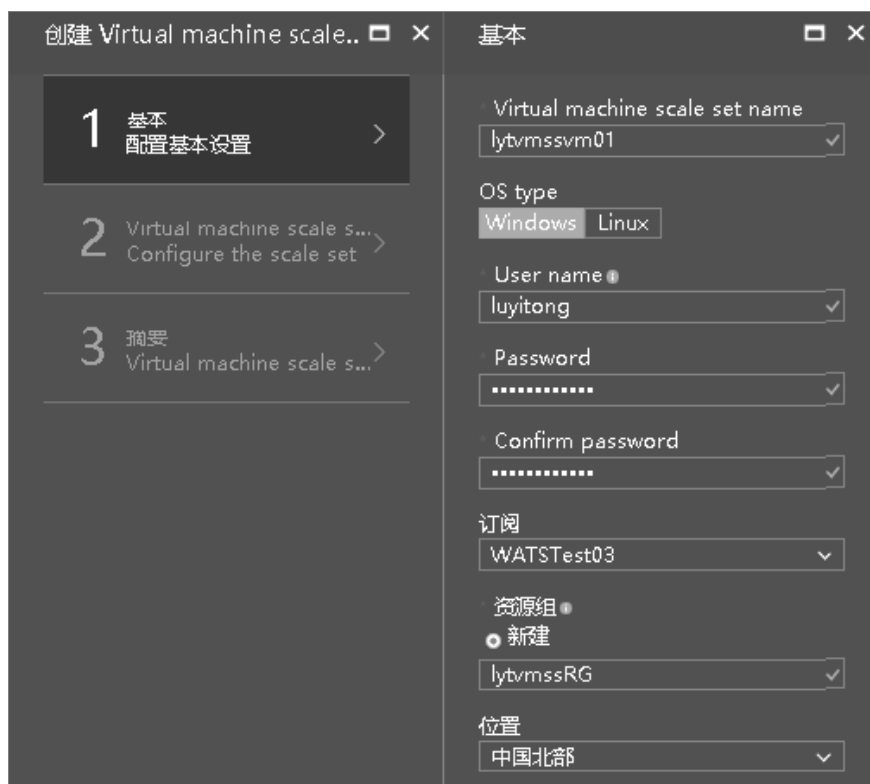


图 14.10-1

(3) 在 Virtual machine scale set service settings 边栏选项卡上：输入所需的域名标签（规模集前端负载均衡器的 FQDN 的基础）。在整个 Azure 中，此标签必须是唯一的。

选择所需的操作系统磁盘映像、实例计数和 VM 大小，如图 14.10-2 所示。

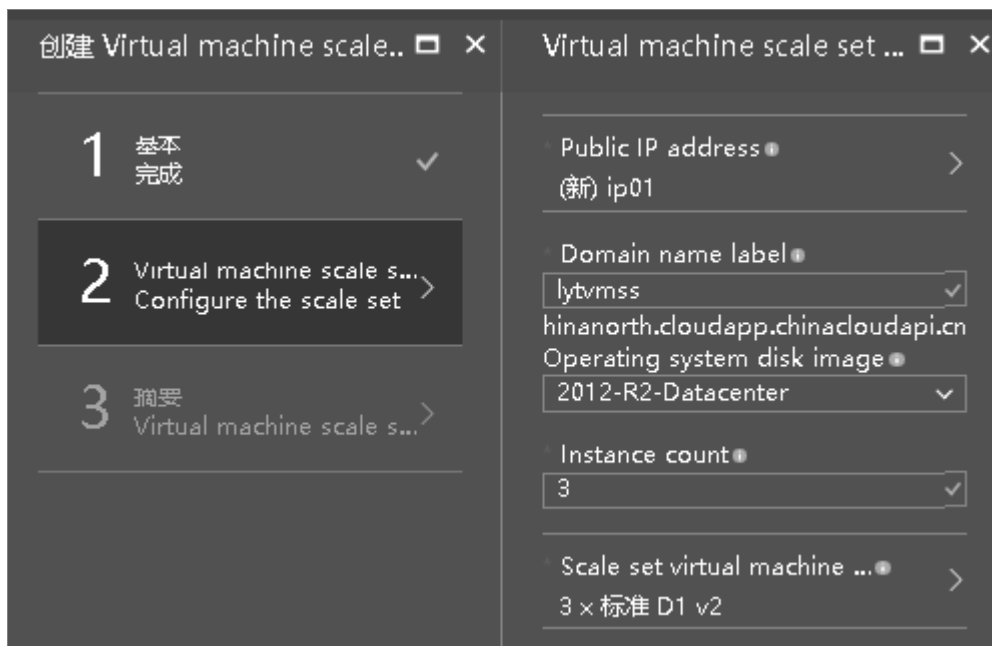


图 14.10-2

(4) 验证完成后，在 **Summary** 边栏选项卡上，单击 **OK** 开始进行规模集部署，如图 14.10-3 所示。



图 14.10-3

(5) 查看部署状态，如图 14.10-4 所示。

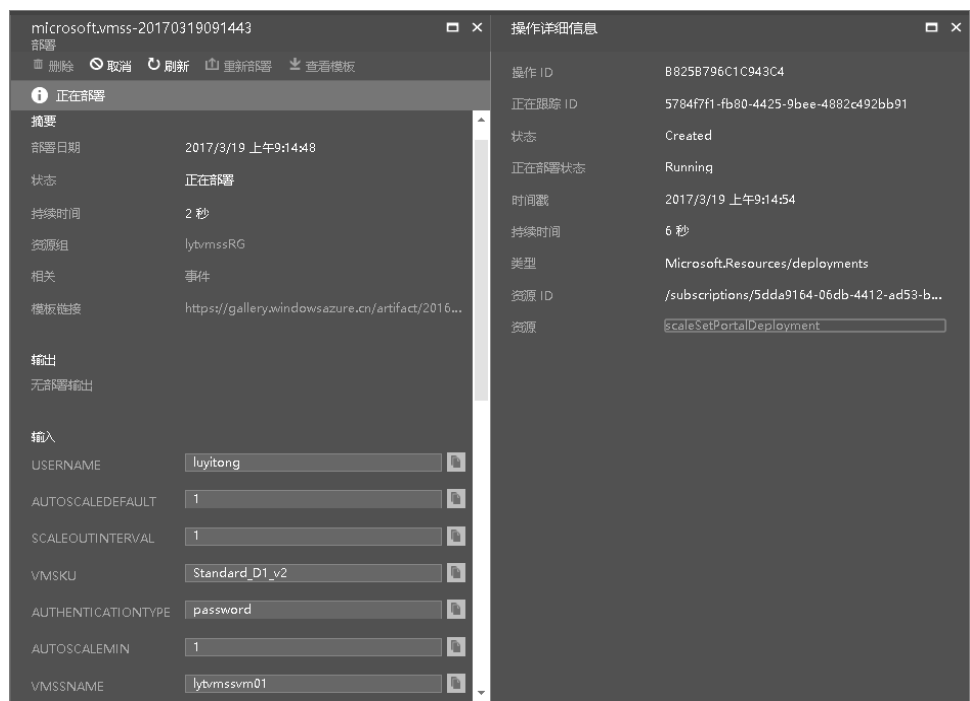


图 14.10-4

(6) 查看该规模集资源组下成功创建的全部资源，如图 14.10-5 所示。



图 14.10-5

**备注：**

- 通过 RDP/SSH 连接到 VM 规模集实例 - VM 规模集是在 VNET 中创建的，并且没有为规模集中单独的 VM 分配公共 IP 地址。这是一件好事，因为您通常不希望承担为计算网格中的所有无状态资源分配单独的公共 IP 地址而产生的支出和管理开销，并且您可以轻松地 VNET 中的其他资源（包括负载均衡器或独立虚拟机等

具有公共 IP 地址的资源) 连接到这些 VM。

- 使用 NAT 规则连接到 VM - 可以创建一个公共 IP 地址, 并将其分配给负载均衡器, 然后定义入站 NAT 池, 用于将 IP 地址上的端口映射到 VM 规模集中的 VM 上的端口。例如:

源	Source Port	目标	Destination Port
公共 IP	端口 50000	vm ss_0	端口 22
公共 IP	端口 50001	vm ss_1	端口 22
公共 IP	端口 50002	vm ss_2	端口 22

### 14.10.2 案例一

**现状:** C 公司是一家电商企业, 用户的访问, 峰值时间都是很难预测的, 尤其是在重要的节假日或商业促销的时候, 传统数据中心到底应该部署什么规模的 Web 集群一直是一个问题, 如果部署过量会造成成本和资源的浪费, 部署过少, 则会在遇到峰值时来不及扩充, 容易造成用户无法访问、用户体验差、交易额损失等等, 同样, 该公司的 IT 运维人员在这个时期就会面临神经紧绷、实时检测的压力情况...

**方案:** VMSS 作为 Azure 云服务新的计算方式, 提供了根据服务压力负载自动扩展收缩, 并且同时能够支持 Windows 和 Linux 系统, 在提供了 IaaS 级别的控制灵活性的同时, 也提供了 PaaS 级别的自动扩展, 对于无状态的 Web 应用服务等场景非常适合。针对 C 公司目前遇到的现状, 该企业 IT Manager 决定将本地数据中心 Web 应用服务迁移到 Azure IaaS 平台, 通过 ARM 模板和 VMSS 创建一个自动负载均衡的, 按照 CPU 负载自动扩展的 Web 服务器集群, 如图 14.10-6 所示。

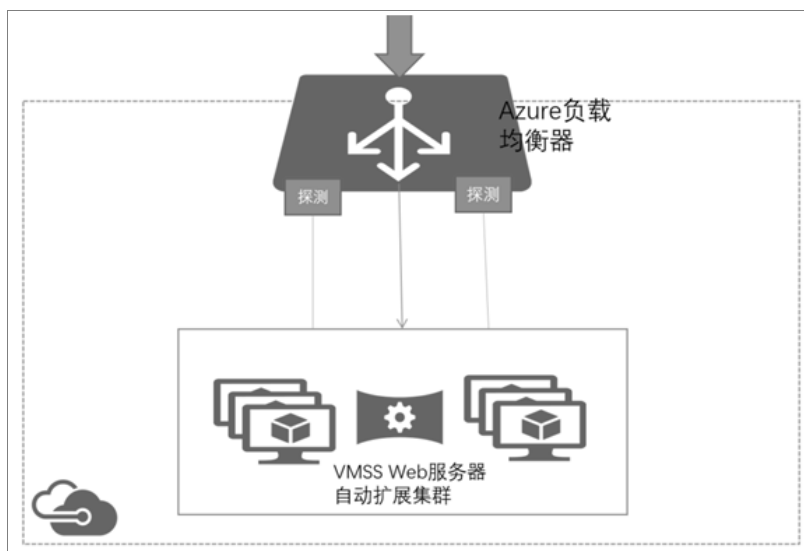


图 14.10-6

在本模板中，将会创建如下实验环境：

- 定义一个负载均衡器，负责转发前端的 Web 请求给后端的 Web 集群
- 使用 VMSS 创建一个 Web 集群
- 使用客户定制化脚本，自动安装 Apache Web 服务器，和 PHP Web 应用
- 定义自动扩展集合的规则，根据虚拟机自动扩展集合中的 CPU 负载进行自动扩展或者收缩，虚拟机也会自动的在负载均衡器中自动添加或者删除
- 压力测试用具，可以使用 LoadRunner，Apache AB 等等，在本例中，使用 PHP 产生压力，达到 CPU 阈值要求

(1) 定义负载均衡器，首先我们需要增加一个负载均衡器的资源，这个资源依赖于公共 IP 地址，即前端 IP 地址。

```
{
  "type": "Microsoft.Network/loadBalancers",
  "name": "[variables('loadBalancerName')]",
  "location": "[resourceGroup().location]",
  "apiVersion": "[variables('networkApiVersion')]",
  "dependsOn": [
    "[concat('Microsoft.Network/publicIPAddresses/',
variables('publicIPAddressName'))]"
  ],
  "properties": {
    "frontendIPConfigurations": [
      {
        "name": "LoadBalancerFrontEnd",
        "properties": {
          "publicIPAddress": {
            "id": "[variables('publicIPAddressID')]"
          }
        }
      }
    ]
  }
}
```

(2) 定义负载均衡规则，前端的请求通过公网 IP 地址或者 DNS 进来，通过默认的地址分发给后端地址池，使用 TCP 协议，前后端均为标准 80 端口号，也可同时设置负载均衡器 idleTime 空闲超时时间，最长可设置为 30 分钟；另外针对 HTTP 请求，需要设置一下针对 80 端口的探测，以此判断后端虚拟机是否健康。

```
{
  "loadBalancingRules": [
    {
      "name": "LBRule",
      "properties": {
        "frontendIPConfiguration": {
          "id": "[variables('frontEndIPConfigID')]"
        },
        "backendAddressPool": {
          "id": "[variables('lbPoolID')]"
        }
      },
      "protocol": "tcp",
      "frontendPort": 80,

```



```

    "backendPort": 80,
    "enableFloatingIP": false,
    "idleTimeoutInMinutes": 5,
    "probe": {
      "id": "[variables('lbProbeID')]"
    },
    .....
    "probes": [
      {
        "name": "tcpProbe",
        "properties": {
          "protocol": "tcp",
          "port": 80,
          "intervalInSeconds": "5",
          "numberOfProbes": "2"
        }
      },
      .....
    ]
  },
  .....

```

(3) Azure 提供了定制化脚本扩展，以方便用户定制化部署，可以让你在虚拟机部署完成后，运行自定义的脚本，安装自己软件和应用，具体的用法如下，你可以将你的应用放在 Azure 存储中，本例中放在了 github 上，然后执行 bash，进行安装配置：

```

    },
    "extensionProfile": {
      "extensions": [
        {
          "name": "lapextension",
          "properties": {
            "publisher": "Microsoft.OSTCEExtensions",
            "type": "CustomScriptForLinux",
            "typeHandlerVersion": "1.4",
            "autoUpgradeMinorVersion": false,
            "settings": {
              "fileUris": [
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/install_lap.sh",
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/index.php",
                "https://raw.githubusercontent.com/kingliantop/azurelabs/master/AzureChinaARMTemplate/vmss-lapstack-autoscale/do_work.php"
              ]
            },
            "commandToExecute": "bash install_lap.sh"
          }
        }
      ]
    }
  }
}

```

本次测试中，提供了两个 PHP Web 文件，一个是 index.php，用来显示当前的 Web 应用跑在哪个服务器上，另外一个 do\_work.php 用来给 Web 服务器产生压力，触发自动扩展。

(4) 配置 VMSS 自动扩展规则，例如在什么情况下自动扩展或自动收缩。在本例中，定义 VMSS 中 CPU 负载在过去的 5 分钟内高于 60% 就进行自动扩展，低于 50% 就自动收缩。

```

    "rules": [
      {
        "metricTrigger": {
          "metricName": "\\Processor\\PercentProcessorTime",
          "metricNamespace": "",
          "metricResourceUri": "[concat('/subscriptions/',
subscription().subscriptionId, '/resourceGroups/', resourceGroup().name,
'/providers/Microsoft.Compute/virtualMachineScaleSets/',
variables('namingInfix'))]",
          "timeGrain": "PT1M",
          "statistic": "Average",
          "timeWindow": "PT5M",
          "timeAggregation": "Average",
          "operator": "GreaterThan",
          "threshold": 60.0
        },
        "scaleAction": {
          "direction": "Increase",
          "type": "ChangeCount",
          "value": "1",
          "cooldown": "PT1M"
        }
      },
      {
        "threshold": 50.0
      },
      {
        "scaleAction": {
          "direction": "Decrease",
          "type": "ChangeCount",
          "value": "1",
          "cooldown": "PT1M"
        }
      }
    ]
  }

```

(5) 配置参数文件，定义 VMSS 的名称、初始在 VMSS 中虚拟机数量、用户名和密码。

```

{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/
deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmssName": {
      "value": "Webscaleset"
    },
    "instanceCount": {
      "value": 2
    },
    "adminUsername": {
      "value": "XXXXXX"
    },
    "adminPassword": {

```

```

    "value" : "XXXXXX"
  }
}

```

(6) 最后, 我们使用 Powershell 进行部署, 如图 14.10-7 所示。

```

PS D:\vmss> .\deploy.ps1
详细信息: 22:33:31 - Created resource group 'lytwebscale' in location 'chinanorth'

ResourceGroupName : lytwebscale
Location           : chinanorth
ProvisioningState   : Succeeded
Tags               :
ResourceId          : /subscriptions/5dda9164-06db-4412-ad53-bb0521e11111/resourceGroups/lytwebscale

New-AzureRmResourceGroupDeployment
-Name
lytwebscale
-ResourceGroupName
lytwebscale
-TemplateFile

```

图 14.10-7

(7) 部署完成后, 登录 Azure Portal: <https://portal.azure.cn>, 可以看到新的 VMSS 集合已经部署成功, 包括有一个扩展集, 一个负载均衡器, 一个公网 IP 地址及多个用于分发 VM 的存储账号, 如图 14.10-8 所示。

名称	类型	位置
部署		中国北部
1 成功		
按名称筛选...		
10 个项		
名称	类型	位置
bqqqdcgatabo2webscalesa	存储帐户	中国北部
h3awlkkj5id6webscalesa	存储帐户	中国北部
t2msvvsd3awpewbscalesa	存储帐户	中国北部
vhph2ej7tp25cwebscalesa	存储帐户	中国北部
webscales	虚拟机规模集	中国北部
webscaleslb	负载均衡器	中国北部
webscalespip	公共 IP 地址	中国北部
webscalesvnet	虚拟网络	中国北部
zghaq554wdocwebscalesa	存储帐户	中国北部

图 14.10-8

(8) 进入虚拟机扩展集, 查看当前实例, 可以看到当前有 2 个实例, 如图 14.10-9 所示。

名称	状态	最新模型
<input type="checkbox"/> webscales_0	正在运行	是
<input type="checkbox"/> webscales_1	正在运行	是

图 14.10-9

(9) 打开负载均衡器, 获得公网的 IP 地址或者 DNS, 在浏览器中打开, 可以看到当前连接的是 001 Web 服务器, 该页面是一个 demo 页面, 用于给虚拟机产生压力; 新打开一个浏览器, 连接负载均衡器, 可以看到请求被分发到了 002 Web 服务器, 如图 14.10-10 所示。



图 14.10-10

(10) 在当前的测试页面上, 输入 500 秒, 作为压力测试时长, 单击 "DO work", 那么 PHP 程序就会产生压力, 占满 CPU, 如图 14.10-11 所示。

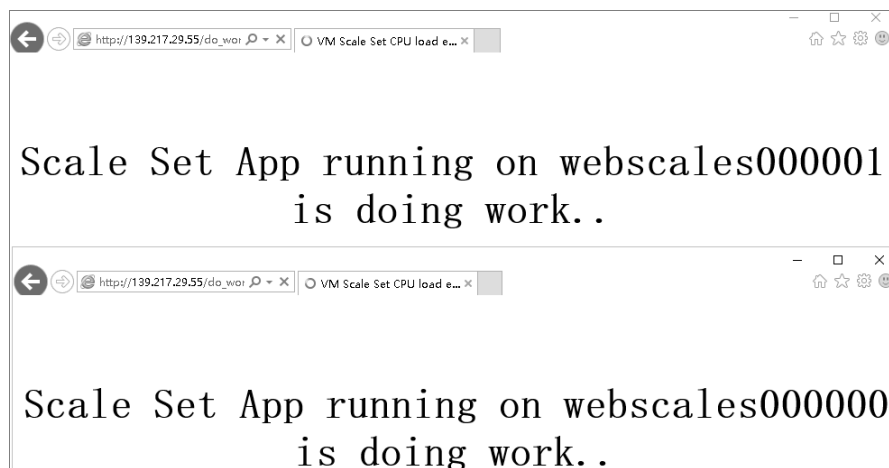


图 14.10-11

(11) CPU 负载连续 5 分钟超过 60% 后, 我们打开虚拟机扩展页面的实例项, 可以看到, 按照之前模板定义的 VMSS 自动扩展规则, 虚拟机开始自动增加, 如图 14.10-12 所示。

搜索虚拟机实例		
名称	状态	最新模型
<input type="checkbox"/> webscales_0	● 正在运行	是
<input type="checkbox"/> webscales_1	● 正在运行	是
<input type="checkbox"/> webscales_4	⦿ 正在创建 (正在运行)	是
<input type="checkbox"/> webscales_5	⦿ 正在创建 (正在运行)	是

图 14.10-12

(12) 压力测试完成，虚拟机扩展集的压力逐步低于 50%，这个时候，整个虚拟机扩展集会监测最近 5 分钟的负载情况，一旦满足收缩要求，就会执行 `cooldown` 的过程，逐步移除 Web 服务器，也会从负载均衡器移除，降低成本，如图 14.10-13 所示。

搜索虚拟机实例		
名称	状态	最新模型
<input type="checkbox"/> webscales_0	● 正在运行	是
<input type="checkbox"/> webscales_1	● 正在运行	是
<input type="checkbox"/> webscales_4	⦿ 正在删除	是

图 14.10-13

通过以上实验方案，该企业可以方便的使用 **VMSS+ARM** 快速的构建自动可扩展的 Web 集群，并且使用定制化脚本部署需要的应用程序。

# 第十五章 排错工具与方法

针对在 Azure 中遇到的各种网络问题，有非常多的网络排查和调试工具可以使用。本章针对网络连通性测试，路由检测以及网络抓包推荐了几款常用的工具，并针对工具的使用方法给出了详细说明。

## 15.1 连通性测试

### 15.1.1 psping 工具的使用

为了防止与 ICMP 有关的攻击，Azure 平台默认是屏蔽 ICMP 报文的，因此，检测网络连通性的 ping 命令默认情况下是失效的（如果确实需要使用 ping 工具进行探测，经典模式下可以为虚拟机配置实例级公网 IP）。

为了探测网络连通性，推荐使用 psping 工具，psping 工具是基于 tcp 协议的三次握手进行指定 IP 的端口探测工具，下载地址：<https://technet.microsoft.com/en-us/sysinternals/jj729731.aspx>

下载后无需安装，将下载得到的 pstools.zip 解压，在解压得到的文件中找到 psping.exe 工具，将该工具拷贝到 C:\Windows\System32 目录下即可。其实除了 psping.exe 工具，pstools 中还包括了很多其他实用的工具，这里就不一一介绍了，读者有兴趣可以根据需要学习使用。

psping 工具的使用也比较简单，打开 cmd 窗口，由于我们已经将 psping.exe 放入了 C:\Windows\System32 下，所以不需要再额外添加环境变量，直接在当前路径中输入 psping 命令即可，这里以百度的 80 端口为例，使用下面的命令进行 psping 探测：

```
C:\Users\XXX>psping -n 6 www.baidu.com:80
```

命令执行后，会对 www.baidu.com 的 80 端口进行 6 次 TCP 探测，并返回每次探测的延时，统计平均/最大/最小延迟以及丢包率，如图 15.1-1 所示。

psping 工具除了上面的 -n 参数外，还有其他一些参数，具体可以使用 psping 命令查看其帮助说明，例如 psping -? t 用于查看 TCP ping 的帮助。

### 15.1.2 tcpping 工具的安装和使用

psping 工具只能用于 Windows，如果要在 Linux 系统中进行基于 TCP 的探测，可以使用 tcpping 工具。下面介绍在 CentOS 中安装 tcpping 的基本步骤。

```

C:\Users\DanielHX>psping -n 6 www.baidu.com:80

PsPing v2.01 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2014 Mark Russinovich
Sysinternals - www.sysinternals.com

TCP connect to 111.13.100.92:80:
7 iterations (warmup 1) connecting test:
Connecting to 111.13.100.92:80 (warmup): 8.64ms
Connecting to 111.13.100.92:80: 8.03ms
Connecting to 111.13.100.92:80: 7.96ms
Connecting to 111.13.100.92:80: 6.29ms
Connecting to 111.13.100.92:80: 6.76ms
Connecting to 111.13.100.92:80: 7.43ms
Connecting to 111.13.100.92:80: 7.49ms

TCP connect statistics for 111.13.100.92:80:
  Sent = 6, Received = 6, Lost = 0 (0% loss),
  Minimum = 6.29ms, Maximum = 8.03ms, Average = 7.33ms

```

图 15.1-1

tcpping 工具是一个第三方的开源工具，基于 tcptraceroute，所以在安装 tcpping 前，需要先安装 tcptraceroute 工具。

CentOS 默认源中未包含 tcptraceroute 包，所以要先添加 repoforge 源，从 repoforge 网站 <http://repoforge.org/use/> 上找到系统的发行版本对应的 repoforge 源的对应版本，由于这里使用的是 CentOS 6.5 64 位系统，所以选择 EL 6 的 x86\_64 的 rpm 包，使用下面两条命令下载并添加 repoforge 源：

```

[root@XXX ~]# wget http://repository.it4i.cz/mirrors/repoforge/redhat/el6/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm
[root@XXX ~]# rpm -ivh rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm

```

添加完成后，使用 yum 命令安装 tcptraceroute 包：

```
[root@XXX ~]# yum install -y tcptraceroute
```

完成安装后，切换到/usr/bin 下，下载 tcpping 工具，并修改其执行权限：

```

[root@XXX ~]# cd /usr/bin
[root@XXX ~]# wget http://www.vdberg.org/~richard/tcpping
[root@XXX ~]# chmod 755 tcpping

```

这里就完成了 tcpping 工具的安装，使用方法与 psping 工具类似，例如：

```

[root@XXX ~]# tcpping www.baidu.com 80
seq 1: tcp response from 220.181.112.244 [open] 35.714 ms
seq 2: tcp response from 220.181.111.188 [open] 32.019 ms
seq 3: tcp response from 220.181.111.188 [open] 32.785 ms
seq 4: tcp response from 220.181.112.244 [open] 32.174 ms

```

```
seq 0: tcp response from 220.181.111.188 [open] 35.916 ms
seq 5: tcp response from 220.181.111.188 [open] 32.642 ms
seq 6: tcp response from 220.181.112.244 [open] 28.894 ms
seq 7: tcp response from 220.181.111.188 [open] 32.277 ms
```

关于 `tcpping` 更详细的参数说明，可以直接输入 `tcpping` 查看其参数说明。

### 15.1.3 paping 工具的使用

除了 `tcpping`，在 Linux 上也可以使用 `paping` 来进行端口探测，当然 `paping` 也可以用于 Windows。下面介绍在 CentOS 中安装并使用 `paping` 进行端口探测的步骤：

从 <http://code.google.com/p/paping> 下载 `paping` 的 tar 包，由于 [code.google.com](http://code.google.com) 是国外网站，如果遇到无法访问的情况，可以在国内一些软件下载网站上查找 `paping` 的下载资源。

下载后使用下面的命令进行解压：

```
[root@XXX ~]# tar zvxf paping_1.5.5_x86-64_linux.tar.gz
```

然后无需安装，使用下面的命令进行端口连通性探测：

```
[root@XXX ~]# ./paping www.baidu.com -p 80 -c 6
paping v1.5.5 - Copyright (c) 2011 Mike Lovell
Connecting to www.a.shifen.com [220.181.112.244] on TCP 80:
Connected to 220.181.112.244: time=35.98ms protocol=TCP port=80
Connected to 220.181.112.244: time=35.94ms protocol=TCP port=80
Connected to 220.181.112.244: time=32.33ms protocol=TCP port=80
Connected to 220.181.112.244: time=35.98ms protocol=TCP port=80
Connected to 220.181.112.244: time=35.76ms protocol=TCP port=80
Connected to 220.181.112.244: time=35.78ms protocol=TCP port=80
Connection statistics:
    Attempted = 6, Connected = 6, Failed = 0 (0.00%)
Approximate connection times:
    Minimum = 32.33ms, Maximum = 35.98ms, Average = 35.29ms
```

### 15.1.4 nc 工具的使用

与上面这些探测工具相比，`nc` 是一款更为复杂和强大的工具，`nc` 是 NetCat 的缩写，安装比较简单，仍然以 CentOS 系统为例，使用 `yum` 可以直接安装：

```
[root@XXX ~]# yum install nc
```

`nc` 的功能有很多，这里举两个使用小例子来做说明：

检测端口连通性：

```
[root@XXX ~]# while true; do nc -z www.baidu.com 80; done > result.txt
^C
[root@XXX ~]# cat result.txt
Connection to www.baidu.com 80 port [tcp/http] succeeded!
Connection to www.baidu.com 80 port [tcp/http] succeeded!
Connection to www.baidu.com 80 port [tcp/http] succeeded!
```



```
Connection to www.baidu.com 80 port [tcp/http] succeeded!
.....
```

端口扫描（时间会比较长）：

```
[root@XXX ~]# nc -z -w 1 www.baidu.com 1-100
Connection to www.baidu.com 80 port [tcp/http] succeeded!
```

关于 nc 更多的参数和使用方法，可以参考公网上的一些资料进行深入学习。

## 15.2 路由检测

### 15.2.1 Windows 中使用 tracert 探测路由

通常在 Windows 中使用 tracert 工具进行路由探测，tracert 是基于 ICMP 报文，利用 IP 头部中的 TTL 字段来进行网络路由路径探测。

tracert 最多会探测 30 跳路由，每一跳默认会探测 3 次，使用方法如下：

```
C:\Users\XXX>tracert www.baidu.com
Tracing route to www.a.shifen.com [111.13.100.92]
over a maximum of 30 hops:
  1    1 ms    2 ms    1 ms  192.168.1.1
  2   19 ms    5 ms    9 ms  10.105.0.1
  3    4 ms    3 ms    5 ms  10.2.1.5
  4    8 ms    6 ms    7 ms  10.10.10.13
  5    7 ms   10 ms    7 ms  10.9.3.1
  6    4 ms    4 ms    4 ms  39.155.188.13
  7    *      *      *      Request timed out.
  8    5 ms    6 ms    5 ms  111.13.0.170
  9   159 ms    6 ms    7 ms  111.13.188.65
 10    8 ms    9 ms    7 ms  111.13.188.81
 11   11 ms    7 ms    7 ms  111.13.112.53
 12    *      *      *      Request timed out.
 13    7 ms    6 ms    7 ms  127.0.0.1 [111.13.100.92]
Trace complete.
```

可以看到，结果中会统计每一跳路由的 3 次探测延迟，如果显示为\*则表示对应的一次探测超时，每一行最右边会显示探测到的路由的 IP 地址。

由于 Azure 是屏蔽 ICMP 报文的，所以对于经典模式下部署的资源而言，想要使用 tracert 向外探测，需要为对应的虚拟机配置实例级公网 IP。

### 15.2.2 Linux 中使用 mtr 探测路由

在 Linux 中可以使用 mtr 工具进行路由探测，mtr 也是使用 ICMP 报文进行探测，所以在 Azure 经典模式中，需要为虚拟机指定实例级公网 IP 才能够正常使用 mtr 工具。

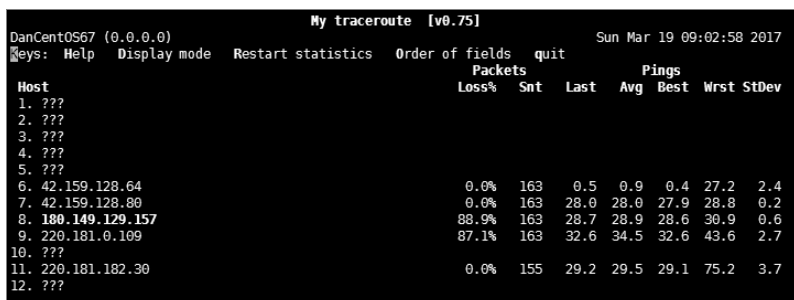
mtr 工具并非系统自带工具，以 CentOS 系统为例，使用 yum 包管理工具进行安装：

```
[root@XXX ~]# yum install -y mtr
```

安装完成后，使用下面的命令检测路由：

```
[root@XXX ~]# mtr www.baidu.com
```

mtr 是一个持续的统计工具，在退出（按 q 键退出）前会持续进行每一跳的探测和统计，如图 15.2-1 所示。



Host	Packets Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. ???							
2. ???							
3. ???							
4. ???							
5. ???							
6. 42.159.128.64	0.0%	163	0.5	0.9	0.4	27.2	2.4
7. 42.159.128.80	0.0%	163	28.0	28.0	27.9	28.8	0.2
8. 180.149.129.157	88.9%	163	28.7	28.9	28.6	30.9	0.6
9. 220.181.0.109	87.1%	163	32.6	34.5	32.6	43.6	2.7
10. ???							
11. 220.181.182.30	0.0%	155	29.2	29.5	29.1	75.2	3.7
12. ???							

图 15.2-1

关于 mtr 的详细用法，可以使用 `man mtr` 查看使用帮助手册。

## 15.3 抓包工具

### 15.3.1 使用 Network Monitor 进行抓包

Network Monitor 是微软推出的一款 Windows 平台的网络排查工具，可以利用其进行网络数据包的抓取和分析。

Network Monitor 的下载地址：<https://www.microsoft.com/en-us/download/details.aspx?id=4865>

安装后，使用管理员权限打开 Network Monitor 工具（如果打开发现检测不到网卡的情况，大多是由于没有以管理员权限运行），如图 15.3-1 所示。

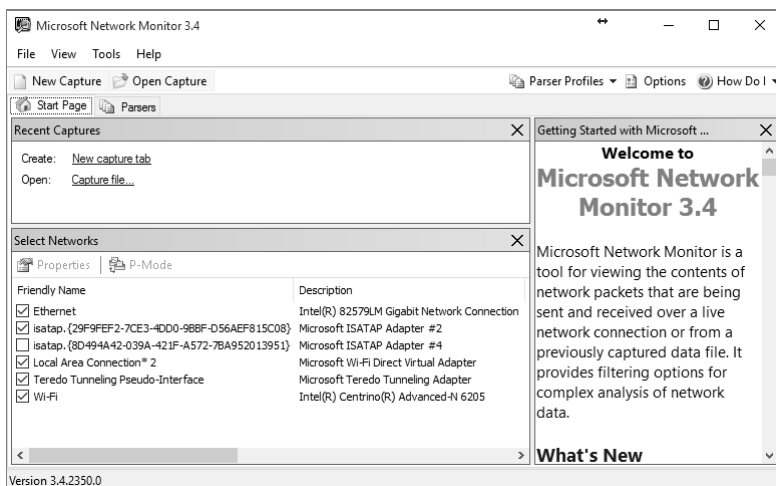


图 15.3-1

单击界面左上角的 New Capture，接着单击上方命令按钮中的“Start”按钮开始抓取数据包，如图 15.3-2 所示。



图 15.3-2

抓取的同时，可以在“Display Filter”中加入自定义的过滤规则，单击“Apply”按钮将规则应用于抓包结果，如图 15.3-3 所示。

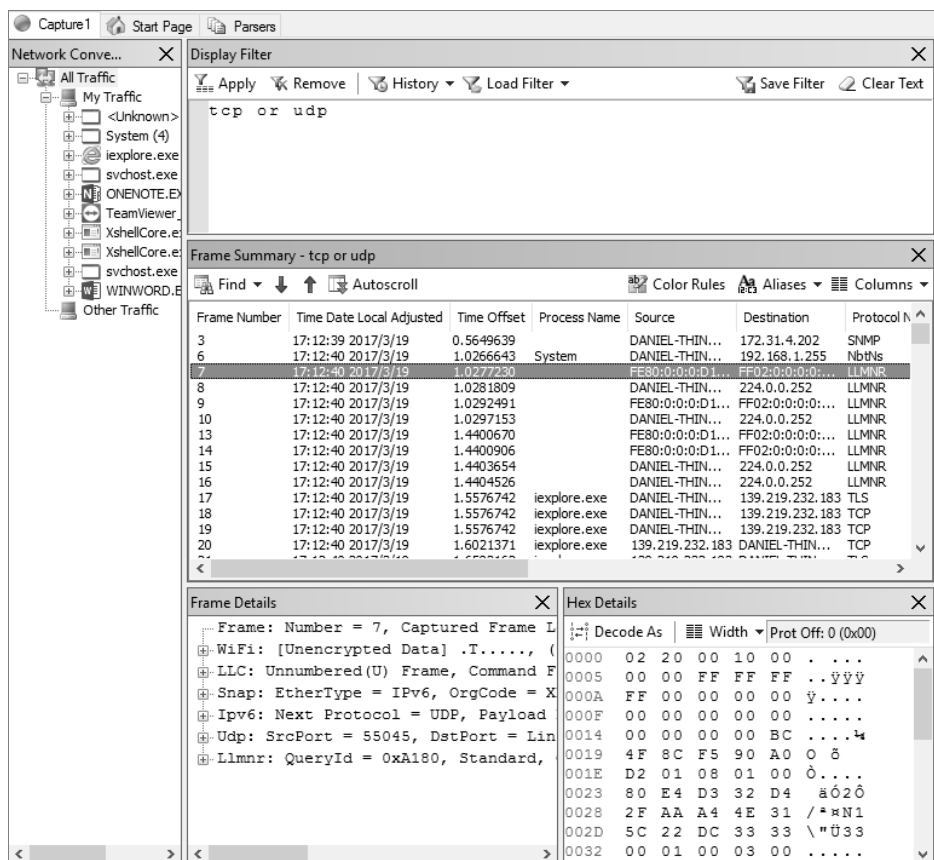


图 15.3-3

左侧窗口中显示了抓取得到的全部数据包的所属进程，选择某个进程可以在右侧将此进程对应的数据包规律出来，还可以展开进程名称左侧的“+”号，按照进程通信的 IP 地址进行进一步过滤。

右侧中间部分“Frame Summary”窗口中显示出根据目前进程，过滤条件筛选得到的数据包概况，选中某个数据包后，可以在下方的“Frame Details”中展开查看报文的具体内容，“Hex Details”对应了报文内容的 16 进制编码。

抓包结束后，单击上方的“Stop”按钮即可停止抓包，可以将抓包结果保存为“.cap”文件留待后续分析。

关于 Network Monitor 更详细的使用方法，可以参考 Network Monitor Blog 中的内容：  
<https://blogs.technet.microsoft.com/netmon/>

### 15.3.2 使用 netsh 进行抓包

在没有安装 Network Monitor 的环境中，例如很多 Windows 服务器安装时为了最大化服务器资源，仅安装了 Server Core，这种情况下，可以使用内置的命令行工具 netsh 进行网络数据包的抓取。

netsh 抓包命令如下：

```
C:\Users\XXX>netsh trace start capture=yes maxSize=200M overwrite=yes
tracefile=c:\nettrace.etl
C:\Users\XXX>netsh trace stop
```

上面的参数指定了最大抓取的文件大小，覆盖选项，输出文件路径。除了这些参数，还可以指定更详细的过滤条件，具体可以参考链接：<https://technet.microsoft.com/en-us/library/dd878517.aspx>

抓取得到的“.etl”文件可以导出到安装了 Network Monitor 的 Windows 系统中进行进一步分析。

### 15.3.3 使用 tcpdump 进行抓包

在 Linux 系统中，可以使用 tcpdump 工具进行数据包的抓取，tcpdump 默认已安装，使用下面的命令开始进行数据包抓取：

```
[root@XXX ~]# tcpdump -i eth0 -w server.cap
```

上面命令中，针对 eth0 网络接口进行抓包，将抓取得到的报文写入当前目录下的 server.cap 文件中。如果默认网络接口非 eth0，可以使用下面的命令查看系统中的网络接口名称：

```
[root@XXX ~]# netstat -in
Kernel Interface table
```

Flg	Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR
	eth0	1500	0	118367	0	0	0	150238	0	0	0
BMRU	lo	65536	0	62	0	0	0	62	0	0	0 LRU

除了上面的命令外，还可以针对 tcpdump 添加很多过滤规则，具体可以参考 tcpdump 的用户手册 `man tcpdump`。